

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО”**

Факультет прикладної математики

Кафедра програмного забезпечення комп’ютерних систем

“На правах рукопису”

УДК 004.9

“До захисту допущено”

Науковий керівник кафедри

_____ І. А. Дичка

“ ____ ” _____ 2019 р.

Магістерська дисертація

на здобуття ступеня магістра

зі спеціальності 121 Інженерія програмного забезпечення

**на тему: “Метод та програмне забезпечення оптимального розміщення
джерел світла в приміщенні”**

Виконала:

студентка II курсу, групи КП-81мп

Глінська Євгенія Михайлівна

Керівник:

Доцент кафедри ПЗКС, к.т.н., доцент,

Сулема Є. С.

Консультант з нормоконтролю:

Доцент кафедри ПЗКС, к.т.н., доцент,

Онай М. В.

Рецензент:

В.о. завідувача кафедри ММСА ІПСА, к.т.н., доцент,

Тимошук О. Л.

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць
інших авторів без відповідних
посилань.

Студентка _____

Київ – 2019 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет прикладної математики
Кафедра програмного забезпечення комп'ютерних систем

Рівень вищої освіти – другий (магістерський)

Спеціальність (спеціалізація) – 121 “Інженерія програмного забезпечення”

ЗАТВЕРДЖУЮ

Науковий керівник кафедри

_____ І.А. Дичка

«___» _____ 2018 р.

ЗАВДАННЯ
на магістерську дисертацію студентці
Глінській Євгенії Михайлівні

1. Тема дисертації: “Метод та програмне забезпечення оптимального розміщення джерел світла в приміщенні”,
науковий керівник дисертації: Сулема Євгенія Станіславівна, к.т.н.,
доцент, затверджені наказом по університету від “13” листопада 2019 р. № 3895-с
2. Термін подання студенткою завершеної дисертації “11” грудня 2019 р.
3. Об’єкт дослідження: процес оптимізації розміщення джерел світла.
4. Предмет дослідження: метод оптимального розміщення джерел світла.
5. Перелік задач, які потрібно вирішити:
 - провести аналіз методів розрахунку рівня освітлення в приміщенні;
 - провести аналіз методів оптимізації;
 - розробити метод оптимального розміщення джерел світла в приміщенні;
 - розробити програмний продукт, який буде оптимізувати розміщення джерел світла з використанням розробленого методу.
6. Орієнтовний перелік графічного матеріалу:
 - архітектура програмного продукту;

- алгоритм оптимізації розміщення джерел світла;
- алгоритм розрахунку рівня освітлення;
- варіанти використання програмного продукту;
- презентація.

7. Орієнтовний перелік публікацій:

- Тези доповіді “Програмний спосіб моделювання оптимального розміщення джерел світла” – XII наукова конференція магістрантів та аспірантів “Прикладна математика та комп’ютинг” (ПМК-2019).

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Онай М. В., к.т.н., доцент		

9. Дата видачі завдання “15” жовтня 2018 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів дисертації	Примітка
1.	Грунтовне ознайомлення з предметною галуззю	14.11.2018	
2.	Визначення структури магістерської дисертації; вивчення літератури, пошук додаткової літератури, патентний пошук	28.12.2018	
3.	Робота над першим розділом магістерської дисертації; проведення наукового дослідження	15.02.2019	
4.	Проведення наукового дослідження; робота над другим розділом; розроблення програмного забезпечення	30.05.2019	
5.	Робота над третім розділом; робота над статтею за результатами наукового дослідження	03.09.2019	
6.	Завершення роботи над програмним забезпеченням; робота над четвертим розділом; робота над статтею за результатами наукового дослідження	17.10.2019	
7.	Завершення роботи над основною частиною магістерської дисертації; робота над п’ятим розділом; підготовка графічного матеріалу	08.11.2019	
8.	Оформлення текстової та графічної частини магістерської дисертації	03.12.2019	

Студентка _____ Є.М. Глінська

Науковий керівник дисертації _____ Є.С. Сулема

РЕФЕРАТ

Актуальність. Комфортність проживання майбутнього мешканця квартири або висока продуктивність у робітників офісів стоїть на першому місці у проектувальників нових приміщень. Питання ергономічності приміщення розділяється на два, а то й більше, підпитання: нормальний рівень освітлення, нормальний рівень шуму, нормальна температура в приміщенні тощо. Тобто це ті питання, що входять в межі норм, задані в стандартах ергономіки приміщення. Перелічені підпитання є найбільш важливими для ергономічного приміщення. В даній дисертації буде розглядатися питання оптимального розміщення джерел світла та його вирішення з метою підвищення ергономічності приміщення, а саме відповідність до норм освітлення.

Актуальність даного питання полягає у відсутності програмних рішень, які дозволили б оптимально розмістити джерела світла в приміщенні відповідно до норм будівельного стандарту ДБН В.2.5-28:2018 “Природне та штучне освітлення”, а також у тому, що невідповідність нормам призводить до погіршення здоров’я людини. Наприклад, в неї розвивається безсоння, дратівливість, сухість очей у випадку надмірного освітлення, або ж зниження гостроти зору у випадку недостатнього рівня світла. Окрім цього, надмірне використання штучного світла призводить до зменшення строку служби ламп та високих затрат на електроенергію. А надмірна кількість природного світла призводить до високої температури в кімнаті та великої кількості відбитих променів світла від поверхонь, яке може викликати значний дискомфорт у людини, яка знаходиться в приміщенні. Саме тому задача оптимального розміщення джерел світла є досить актуальною.

Об’єктом дослідження є процес оптимізації розміщення джерел світла.

Предметом дослідження є способи та алгоритми оптимального розміщення джерел світла.

Метою дослідження є створення програмного забезпечення, яке дозволить оптимально розмістити джерела світла з метою підвищення ергономічності приміщення в сенсі світла та зниження споживання електроенергії.

Методи дослідження. В роботі використовуються методи програмної інженерії, математичного моделювання та емпіричні методи.

Наукова новизна роботи полягає у розробленні методу оптимального розміщення джерел світла, який дозволяє оптимізувати освітлення приміщень за допомогою джерел штучного та природного світла. Суть методу полягає у розбитті площі приміщення на квадрати, площа яких залежить від потужності обраного штучного джерела світла або кількості та розміру вікон бокового освітлення, які розраховуються на основі обчислення рівня освітлення приміщення за рівнем КПО. За тим вікна розташовуються рівномірно по зовнішніх стінах приміщення. На відміну від інших методів оптимізації цей метод не потребує побудови відповідних математичних моделей, формування великої кількості критеріїв та обмежень.

Практична цінність отриманих в роботі результатів полягає у тому, що запропонований спосіб надає можливість перевірити початковий рівень освітлення на відповідність будівельним нормам та провести оптимізацію розміщення джерел світла.

Апробація роботи. Основні положення і результати роботи доповідалися та обговорювалися на XII науковій конференції магістрантів та аспірантів “Прикладна математика та комп’ютинг” ПМК-2019.

Структура та обсяг роботи. Магістерська дисертація складається зі вступу, п’яти розділів, висновків та додатків.

У вступі надано мету дослідження, виконано оцінку сучасного стану проблеми та обґрунтовано актуальність напрямку дослідження.

У першому розділі розглянуто існуючі типи світла, загальні підходи для розрахунку рівня освітлення та їх порівняння між собою.

У другому розділі сформульовано науково-інноваційну задачу, яка полягає у розробленні методу оптимального розміщення джерел світла та його програмної реалізації, проведено аналіз стандарту, в якому наведені норми освітлення для кожного з типів приміщення, проаналізовано основні методи оптимізації та на основі цього сформульовано метод оптимізації розміщення світильників в приміщенні, а також вікон.

У третьому розділі сформульовані основні функціональні та нефункціональні вимоги до програмного забезпечення, проведено аналіз та порівняння 3D-редакторів, а також наведена структура продукту та його основні алгоритми.

У четвертому розділі наведена методика тестування, в якій розроблена послідовність дій для тестування інноваційного програмного продукту на відповідність вимогам та на працездатність та створені набори тестових даних та тестові сценарії, проаналізовано результати та зроблено висновки.

У п'ятому розділі з метою оцінки комерційної життєздатності програмної реалізації розробленого способу було розглянуто можливість створення стартап-проекту на її основі. Розглянуто сучасні проблеми, виявлено зацікавлені сторони та конкурентні переваги, а також створено бізнес-модель і наведено показники потенційної прибутковості.

У висновках проаналізовано результати роботи.

У додатках наведено фрагменти вихідного коду програмної реалізації способу та копії графічних матеріалів.

Робота виконана на 89 аркушах, містить 2 додатки та посилання на список використаних літературних джерел з 15 найменувань. У роботі наведено 12 рисунків та 11 таблиць.

Ключові слова: освітлення, Blender, рівень освітлення, оптимізація, 3D-моделювання.

ABSTRACT

Relevance. The comfort of living of the future tenant of the apartment or high productivity of office workers is the priority of designers of new premises. The issue of room ergonomics is divided into two, or even more, sub-questions: normal lighting level, normal noise level, normal room temperature, etc. These are issues that included in the limits of norms in the room's ergonomics standards. The listed sub-questions are the most important for an ergonomic room. This dissertation will address the issue of the optimal placement of light sources and its solutions in order to increase the ergonomics of the room, according with lighting standards.

The relevance of this issue concluded in the lack of software solutions that would optimally place the light sources in the room in accordance with the norms of the building standard DBN B.2.5-28: 2018 "Natural and artificial lighting", and also that inconsistency with the standards leads to deterioration of human health. For example, a person develops insomnia, irritability, dry eyes in case of excessive lighting, or a decrease in visual acuity in case of insufficient light level. In addition, the excessive use of artificial light leads to a reduction of lamp's lifetime and high electricity expenses. In addition, an excessive amount of natural light leads to a high temperature in the room and a large number of rays of light reflected from the surfaces, which can cause significant discomfort for a person in the room. That is why the task of optimal placement of light sources is very important.

The object of research is the process of optimizing the placement of light sources.

The subject of research is the methods and algorithms of optimal placement of light sources.

The aim of research is to develop the software that allows to place light sources optimally in order to increase the ergonomics of the room in the sense of light and reduce energy consumption.

Research Methods. The paper uses methods of software engineering, mathematical modeling and empirical methods.

The scientific novelty of the work consists in developing a method of optimal placement of light sources, which allows to optimize the illumination of premises with the help of sources of artificial and natural light. The essence of the method consists in dividing the area of the room into squares, which sizes depends on the power of the selected artificial light source or the number and size of the sidewall windows, calculated on the basis of the level of illumination of the room by the KPO level. The windows then are placed evenly on the outer walls of the room. Unlike other optimization methods, this method does not require the construction of appropriate mathematical models and the formation of a huge number of criteria and constraints.

The practical value of the results that obtained in the work consists in the fact that the proposed method allows you to check the initial level of lighting for conformity with construction standards and to optimize the placement of light sources.

Approbation of work. The main conditions and results of the work were reported and discussed at the XII scientific conference of undergraduates and graduate students “Applied Mathematics and Computing” AMC-2019.

The structure and scope of work. The master's thesis consists of introduction, five section, conclusions and appendices.

The introduction presents the purpose of the research, evaluated the current state of the problem and justified the relevance of the research direction.

The first section considered the existing types of light, general approaches for calculating illumination and their comparison.

The second section formulated a scientific and innovative problem is which consists in developing a method for the optimal placement of light sources and its software implementation, made an analysis of the standard, which shows the lighting standards for each type of room, analyzed the main optimization methods

and, based on this, formulated an optimization method of placement of lamps in the room, as well as windows.

The third section formulated the basic functional and non-functional requirements for software, analyzed and compared 3D editors, and also gave the structure of the product and its basic algorithms.

The fourth section provided a testing methodology in which a sequence of actions is developed for testing an innovative software product for conformity with requirements and for operability, created test data sets and test cases, the results are analyzed, and made the conclusions.

The fifth section considered the possibility of creating a startup project based on the software implementation of the developed method, to evaluate it's commercial viability. Also was considered modern problems identified stakeholders and competitive advantages, created a business model and presented the indicators of potential profitability.

The conclusions analyzed the results of the work.

The appendices show fragments of the source code for the software implementation of the method and copies of graphic materials.

The work was performed on 89 pages, contains 2 applications and links to a list of used literature from 15 items. The work contains 12 figures and 11 tables.

Keywords: lighting, Blender, lighting level, optimization, 3D-modeling.

РЕФЕРАТ

Актуальность. Комфортность проживания будущего жильца квартиры или высокая производительность у офисных работников стоит на первом месте у проектировщиков новых помещений. Вопрос эргономичности помещения разделяется на два, а то и больше, подвопроса: нормальный уровень освещения, нормальный уровень шума, нормальная температура в помещении и т. п. То есть это те вопросы, что входят в пределы норм, заданные в стандартах эргономики помещения. Перечисленные подвопросы являются наиболее важными для эргономичного помещения. В данной диссертации будет рассматриваться вопрос оптимального размещения источников света и его решения с целью повышения эргономичности помещения, а именно соответствие с нормами освещения.

Актуальность данного вопроса заключается в отсутствии программных решений, которые позволили бы оптимально разместить источники света в помещении в соответствии с нормами строительного стандарта ДБН В.2.5-28: 2018 «Естественное и искусственное освещение», а также в том, что несоответствие нормам приводит к ухудшению здоровья человека. Например, у него развивается бессонница, раздражительность, сухость глаз в случае чрезмерного освещения, или снижение остроты зрения в случае недостаточного уровня света. Кроме этого, чрезмерное использование искусственного света приводит к уменьшению срока службы ламп и высоким затратам на электроэнергию. А избыточное количество естественного света приводит к высокой температуре в комнате и большому количеству отраженных лучей света от поверхностей, которое может вызвать существенный дискомфорт у человека, находящегося в помещении. Именно поэтому задача оптимального размещения источников света является весьма актуальной.

Объектом исследования является процесс оптимизации размещения источников света.

Предметом исследования являются способы и алгоритмы оптимального размещения источников света.

Целью исследования является создание программного обеспечения, которое позволит оптимально разместить источники света с целью повышения эргономичности помещения в смысле света и снижение потребления электроэнергии.

Методы исследования. В работе используются методы программной инженерии, математического моделирования и эмпирические методы.

Научная новизна работы заключается в разработке метода оптимального размещения источников света, который позволяет оптимизировать освещение помещений с помощью источников искусственного и естественного света. Суть метода заключается в разбиении площади помещения на квадраты, площадь которых зависит от мощности выбранного искусственного источника света или количества и размера окон бокового освещения, которые рассчитываются на основе вычисления освещенности помещения по уровню КПО. Потом окна располагаются равномерно по наружным стенам помещения. В отличие от других методов оптимизации этот метод не требует построения соответствующих математических моделей, формирования большого количества критериев и ограничений.

Практическая ценность полученных в работе результатов заключается в том, что предложенный способ позволяет проверить начальный уровень освещения на соответствие строительным нормам и провести оптимизацию размещения источников света.

Апробация работы. Основные положения и результаты работы докладывались и обсуждались на XII научной конференции магистрантов и аспирантов «Прикладная математика и компьютеринг» ПМК-2019.

Структура и объем работы. Магистерская диссертация состоит из введения, пяти разделов, заключения и приложений.

Во введении представлена цель исследования, выполнена оценка современного состояния проблемы и обоснована актуальность направления исследования.

В первом разделе рассмотрены существующие типы света, общие подходы для расчета освещенности и их сравнение между собой.

Во втором разделе сформулирована научно-инновационная задача, которая заключается в разработке метода оптимального размещения источников света и его программной реализации, проведен анализ стандарта, в котором приведены нормы освещения для каждого из типов помещения, проанализированы основные методы оптимизации и на основе этого сформулирован метод оптимизации размещения светильников в помещении, а также окон.

В третьем разделе сформулированы основные функциональные и нефункциональные требования к программному обеспечению, проведен анализ и сравнение 3D-редакторов, а также приведена структура продукта и его основные алгоритмы.

В четвертом разделе приведена методика тестирования, в которой разработана последовательность действий для тестирования инновационного программного продукта на соответствие требованиям и на работоспособность и созданы наборы тестовых данных, проанализированы результаты и сделаны выводы.

В пятом разделе для оценки коммерческой жизнеспособности программной реализации разработанного способа была рассмотрена возможность создания стартап-проекта на ее основе. Рассмотрены современные проблемы, выявлены заинтересованные стороны и конкурентные преимущества, а также создана бизнес-модель и приведены показатели потенциальной доходности.

В выводах проанализированы результаты работы.

В приложениях приведены фрагменты исходного кода программной реализации способа и копии графических материалов.

Работа выполнена на 89 листах, содержит 2 приложения и ссылки на список использованных литературных источников из 15 наименований. В работе приведены 12 рисунков и 11 таблиц.

Ключевые слова: освещение, Blender, уровень освещения, оптимизация, 3D-моделирование.

ЗМІСТ

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ.....	4
ВСТУП.....	7
1. АНАЛІЗ МЕТОДІВ РОЗРАХУНКУ РІВНЯ ОСВІТЛЕННЯ В ПРИМІЩЕННІ.....	9
1.1. Метод люмену.....	11
1.2. Метод Сампнера.....	13
1.3. Поточковий метод.....	14
1.4. Узагальнення та порівняльний аналіз існуючих методів.....	15
1.5. Висновки до розділу 1.....	16
2. РОЗРОБЛЕННЯ СПОСОБУ РОЗМІЩЕННЯ ДЖЕРЕЛ ОСВІТЛЕННЯ.....	18
2.1. Постановка науково-інноваційної задачі.....	18
2.2. Формулювання гіпотез.....	18
2.3. Аналіз стандартів освітлення приміщень.....	18
2.4. Аналіз методів оптимізації.....	25
2.5. Формулювання способу оптимального розміщення джерел освітлення.....	27
2.6. Висновки до розділу 2.....	30
3. РОЗРОБЛЕННЯ ІННОВАЦІЙНОГО ПРОГРАМНОГО ПРОДУКТУ.....	32
3.1. Вимоги до інноваційного програмного продукту.....	32
3.2. Вибір засобів розроблення.....	34
3.3. Структура програмного продукту.....	54
3.4. Алгоритм введення даних.....	57
3.5. Алгоритм розрахунку рівня освітлення.....	59
3.6. Алгоритм оптимального розміщення джерел світла.....	61
3.7. Висновки до розділу 3.....	62
4. ТЕСТУВАННЯ ІННОВАЦІЙНОГО ПРОГРАМНОГО ПРОДУКТУ ТА АНАЛІЗ РОЗРОБЛЕНОГО СПОСОБУ.....	63
4.1. Методика тестування.....	63
4.2. Створення тестових моделей приміщення.....	65

4.3. Створення тестових сценаріїв.....	66
4.4. Аналіз отриманих результатів.....	68
4.5. Висновки до розділу 4.....	69
5. ІННОВАЦІЙНІСТЬ ПРОГРАМНОГО ПРОДУКТУ.....	70
5.1. Актуальність проблеми, що вирішує програмний продукт.....	70
5.2. Зацікавлені сторони.....	71
5.3. Основні характеристики програмного рішення.....	73
5.4. Конкурентні переваги рішення.....	74
5.5. Клієнти. Сегменти ринку споживання.....	74
5.6. Унікальна ціннісна пропозиція.....	75
5.7. Доходи та витрати.....	76
5.8. Бізнес-модель.....	77
5.9. Висновки до розділу 5.....	80
ВИСНОВКИ.....	81
СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ.....	87
ДОДАТКИ.....	89

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

Геометричний коефіцієнт природної освітленості – відношення площі ортогональної проекції на робочу площину ділянки умовної небесної півсфери, видимої з розрахункової точки через незаповнений світлопроріз або його частину, від якої розраховується освітленість, до площі основи небесної півсфери:

- доля світла неба в коефіцієнті природної освітленості, а у разі розрахунку геометричного коефіцієнта природної освітленості від протилежного будинку – відношення площі ортогональної проекції на робочу площину ділянки небесної півсфери, що затінюється будинком у розрахунковій точці, до площі основи небесної півсфери;
- доля зовнішнього відбивання в коефіцієнті природної освітленості.

ДБН – Державні Будівельні Норми.

Загальне освітлення – освітлення, за якого світильники розміщуються рівномірно у верхній зоні приміщення (загальне рівномірне освітлення) або локалізовано відносно розміщення обладнання (загальне локалізоване освітлення).

Зональна порожнина – простір зони, наприклад стелі, полу або самої кімнати.

Зоровий дискомфорт – відчуття незручності або напруження, що виникає при незадовільному розподілу яскравості в освітленому просторі, що призводить до відволікання уваги, зниження зосередженості, зорової і загальної стомлюваності.

Коефіцієнт корисної дії (ККД) світильника – відношення повного світлового потоку світильника, виміряного за внормованих умов експлуатування з його власними лампами і пристроями, до суми світлових потоків окремих його ламп, виміряних поза світильником за внормованих умов з тими самими (пускорегулювальними, вимірювальними тощо) пристроями.

Коефіцієнт природної освітленості – відношення освітлення, що утворюється в точці на заданій площині світлом, одержаним безпосередньо

або опосередковано від неба, до одночасного освітлення на горизонтальній площині внаслідок освітлення всією півсферою небосхилу, внесок прямого сонячного світла в утворення цих освітлень вилучають.

Нерівномірність природного освітлення – відношення середнього значення до найменшого значення КПО в межах характерного розрізу приміщення.

Об'єкт розрізнення – предмет, що розглядається, окрема його частина або дефект, які треба розрізнити в процесі роботи.

Освітлювальний прилад – пристрій, який перерозподіляє, фільтрує чи перетворює світловий потік, що випромінюється одним, кількома чи багатьма джерелами світла; містить усі необхідні деталі для кріплення і захисту джерел світла, а також для їх підключення до мережі живлення.

Освітлювальні прилади поділяються на світильники (ближньої дії) і прожектори (дальньої дії).

Показник засліпленості – величина, похідна від коефіцієнта засліпленості.

Показник зорового дискомфорту – критерій оцінки дискомфортної блискавості, яка викликає неприємні відчуття при нерівномірному розподіленні яскравості в полі зору.

Пороговий приріст яскравості – параметр, що регламентує осліплювальну дію освітлювальної установки на людину в полі зору водія транспортного засобу.

Робоча поверхня (базова поверхня) – поверхня, на якій виконується робота і нормується або вимірюється освітленість.

САПР – Система Автоматизованого Проектування.

Світлове середовище – сукупність ультрафіолетових, видимих інфрачервоних випромінювань джерел природного і штучного світла; важлива складова життєвого середовища організмів і рослин, яка визначається світловими потоками джерел світла, що змінюються в результаті взаємодії з навколишнім предметним середовищем; сприймається зором за розподілом світла в просторі.

Характерний розріз приміщення – поперечний розріз, як правило, по середині приміщення, площа якого перпендикулярна до площини

заскленних світлових прорізів (при боковому освітленні) або до поздовжньої осі прогонів приміщення. До характерного розрізу приміщення повинні входити ділянки з найбільшою кількістю робочих місць, а також точки робочої зони, найбільш віддалені від світлових прорізів.

.NET – програмна технологія від компанії Microsoft для створення як звичайних програм, так й веб-застосунків.

API – набір способів, за допомогою яких одна програма може взаємодіяти з іншою (Application Programming Interface).

BIM – Building Information Modeling.

CAD – Computer Aided Design.

CFD – CryptoForge Document.

CSV – Comma-Separated Values.

CU – коефіцієнт утилізації.

DGN – (Design) формат для програм САПР.

HVAC (OBK) – сукупність інженерних систем, метою яких є забезпечення оптимальних умов мікроклімату (heating, ventilation and conditioning).

IFC – Industry Foundation Classes.

LLF – параметр втрати світла.

MEL – Maya Embedded Language.

OSL – Open Shading Language.

OpenVDB – Open Volumetric DataBase.

SDK – набір утиліт, за допомогою яких можна розробити власний продукт для інтеграції з іншою програмою (Software Development Kit).

SLL – Society of Light and Lighting.

SVG – Scalable Vector Graphics.

UMG – Unreal Motion Graphics.

UV – координати на текстурі (U, V).

VFX – Visual Effects.

WYSIWYG – What You See Is What You Get.

ВСТУП

Комфортність проживання майбутнього мешканця квартири або висока продуктивність у робітників офісів стоїть на першому місці у проектувальників нових приміщень. Питання ергономічності приміщення розділяється двома, а то й більше, підпитаннями: нормальний рівень освітлення, нормальний рівень шуму, нормальна температура в приміщенні тощо. Тобто це ті приміщення, що входять в межі норм, задані в стандартах про ергономіку. Перелічені підпитання є найбільш важливими для ергономічного приміщення.

Актуальність питання оптимального розміщення джерел світла та його вирішення з метою підвищення ергономічності приміщення, а саме відповідність до норм освітлення, полягає у відсутності програмних рішень, які дозволили б оптимально розмістити джерела світла в приміщенні відповідно до нормативних показників освітленості приміщень нових та підлягаючих реконструкції будівель і споруд житлового і цивільного призначення, відповідно до норм будівельного стандарту ДБН В.2.5-28:2018 “Природне та штучне освітлення”.

Також є доведеним фактом те, що невідповідність нормам призводить до погіршення здоров'я людини, в неї, у випадку надмірного освітлення, може розвиватися безсоння, дратівливість, сухість очей, або ж, у випадку недостатнього рівня світла, може статися зниження гостроти зору.

Слід також зазначити, що надмірне використання штучного світла призводить до зменшення строку служби ламп та високі затрати на електроенергію, а надмірна кількість природного світла призводить до високої температури в кімнаті та великої кількості відбитих променів від поверхні стін, стелі та підлоги, яке може викликати значний дискомфорт у людини, яка знаходиться в приміщенні.

Оскільки усі вищенаведені фактори роблять питання оптимального розміщення джерел світла досить актуальним, в даній дисертації буде

розглядатися це питання та його вирішення з метою підвищення ергономічності приміщення, а саме відповідність останнього до норм освітлення.

Метою дослідження є підвищення ергономічності приміщення в сенсі світла та зниження споживання електроенергії.

1. АНАЛІЗ МЕТОДІВ РОЗРАХУНКУ РІВНЯ ОСВІТЛЕННЯ В ПРИМІЩЕННІ

Для розв'язання проблеми оптимального розміщення джерел світла всередині приміщення необхідно спочатку розрахувати рівень освітлення в ньому. Враховуючи те, що норми освітлення для комфортного проживання або роботи вже встановлені, є можливість порівняти отримані результати. Рекомендовані норми освітлення наведені в книжках The SLL Code for Lighting [1], написаній Пітером Рейнманом разом із Пітером Бойсом та Джоном Фітзпатріком та The SLL Lighting Handbook [2], написаній тими ж авторами.

Для початку розглянемо види освітлення та методи розрахунку його рівня.

Освітлення приміщень буває таких видів:

Природне освітлення – освітлення приміщень світлом неба (прямим або відбитим), яке проходить крізь світлові прорізи в зовнішніх загороджувальних конструкціях. Воно поділяється на:

- бокове – природне освітлення приміщень крізь світлові прорізи у зовнішніх стінах;
- верхнє – природне освітлення приміщень крізь ліхтарі, світлові прорізи в стінах у місцях перепаду висот будівлі;
- комбіноване – поєднання верхнього і бокового природного освітлення.

Штучне освітлення – освітлення приміщень штучними джерелами світла які мають достатню світлову віддачу та прийнятний термін експлуатації. Воно поділяється на:

- робоче – освітлення, яке забезпечує нормовані умови освітлення (освітленість, якість освітлення) в приміщеннях і в місцях виконання робіт поза будівлями;
- комбіноване – штучне освітлення, яке застосовується для створення досить високих рівнів освітленості на робочих

поверхнях завдяки одночасному використанню загального освітлення та місцевого;

- локалізоване – призначене для окремих зон освітлення з підвищеним рівнем освітленості в певних місцях, наприклад, таких, де виконують роботу;
- місцеве освітлення – освітлення для специфічної зорової задачі на додаток до/і контрольоване окремо від загального освітлення аварійне.

Методів для розрахунку рівня освітлення в приміщенні існує досить багато. Необхідно зазначити, що розрахунки можна проводити для прямого освітлення, непрямого (віддзеркалене світло), розсіюваного та комбінованого.

Існує два основних типи світла, для якого проводять розрахунки: прямий та непрямий.

Пряме світло – це таке світло, що прямує від джерела та досягає поверхні, не відбиваючись ні від чого, та поглинається нею.

Непряме світло – це світло, що прямує від джерела та досягає поверхні, але на відміну від прямого світла, відбивається від неї. Таке світло поділяється на два типи за відбиттям: віддзеркалене та дифузне (або розсіяне відбиття).

Віддзеркалене світло – це світло, що прямує від джерела, досягає деякої поверхні та відбивається від неї під деяким кутом. Необхідно зазначити, що відбивається від поверхні лише в одному напрямку.

Дифузне світло – майже аналогічне дзеркальному, однак різниця в кількості відбитті від поверхні. Якщо при віддзеркаленому світлі промінь відбивається лише один раз та під деяким кутом, то при розсіюваному він відбивається в різних напрямках під різними кутами.

Нижче наведено зображення (рис. 1) відмінності між віддзеркаленим та дифузним світлом.

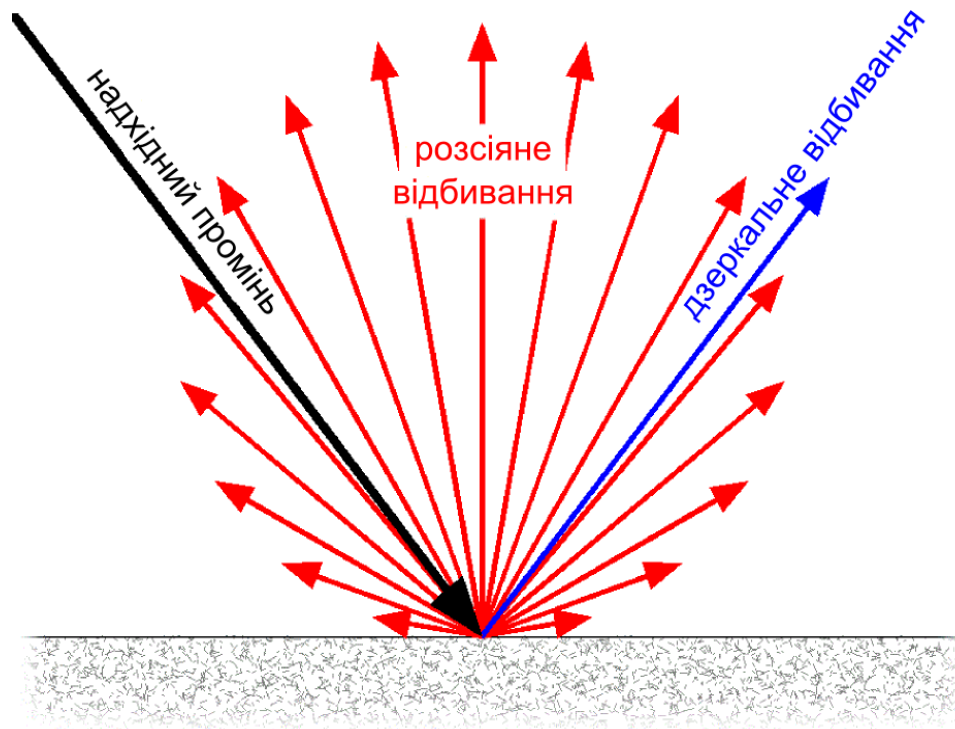


Рис. 1. Відмінність між дифузним та віддзеркаленим світлом

В наступних підрозділах детально розглянуті наступні методи розрахунку освітленості: метод люмену (The lumen method), метод Сампнера (Sumpner's method) та поточковий метод (point-by-point method).

1.1. Метод люмену

Метод люмену [3], або метод “зональної порожнини” (zonal cavity method) є спрощеним методом для обчислення рівня освітлення в приміщенні. Враховує прямий та непрямий потоки світла. Задається наступною формулою:

$$E = \frac{N \cdot n \cdot F \cdot CU \cdot LLF}{A}, \quad (1.1)$$

де N – кількість світильників;

n – кількість ламп в кожному світильнику;

F – світловий потік лампи (в люменах);

CU – коефіцієнт утилізації;

LLF – параметр втрати світла (light loss factor);

A – площа поверхні, на яку падає світло.

Параметр втрати світла (LLF) – це результат множення усіх факторів, що впливають на досягнення світла поверхні. Існують два типа таких факторів: ті, що підлягають відновленню, та ті, що не можуть бути відновлені.

Фактори, що не підлягають відновленню (по-іншому, початкові або фіксовані), визначають частку тієї втрати світла, яке не може бути поновлено шляхом очищення лампи або заміни її елементів. Підходить для ламп, що працюють на іншому світловому значенні, ніж заявлено виробником. Не залежить від віку системи світла.

Такими факторами є:

- температура лампи;
- тепловий коефіцієнт відведення тепла (для ламп, що інтегровані у систему вентиляції (HVAC) ;
- фактор “вольтаж-до-світильника”;
- фактор баласту;
- позиція ламп;
- знос поверхні світильника.

Фактори, що можуть бути поновлені, визначають частку втрати світла, яке може бути відновлено шляхом заміни елементів лампи або очищення їх. Складаються з компонентів, що враховують знос системи освітлення. Використовуються для обчислень для систем, що працюють багато років.

Такими факторами є:

- зношеність лампи;
- накопичення бруду на світильнику;
- накопичення бруду на поверхні в кімнаті;
- вигорання лампи.

Коефіцієнт утилізації (CU, або UF) визначає частку світла, що досягає поверхні, від джерела світла та відбитого від інших поверхонь.

Таким чином, кількість світла, що породжують лампи, помножені на даний коефіцієнт, визначають кількість світла, що досягло поверхні.

Коефіцієнт утилізації [3] можна визначити як відношення світлового потоку, який досяг поверхні, до загального потоку, який випромінюють самі лампи.

Існує вже розраховані табличні значення даного коефіцієнту для типових світильників в книзі IESNA Lighting Handbook [4].

Цей метод також дозволяє знайти необхідну кількість світильників, знаючи рівень освітлення приміщення.

1.2. Метод Сампнера

Даний метод підходить для обчислення як для прямого, так й непрямого освітлення. Цей метод базується на тому, що все світло, яке входить в простір, має або покинути його, або бути поглинутим однією з поверхонь простору [1]. Задається наступною формулою:

$$E = \frac{\Phi}{A (1 - R_{av})}, \quad (1.2)$$

де Φ – кількість світлового потоку, що увійшов у простір;

A – площа поверхні;

R_{av} – зважена за площею середня дзеркальність поверхні, отже $1 - R_{av}$

дає нам її середню поглинальність.

Для обчислення непрямого освітлення використовують наступну формулу:

$$E_{ind} = \frac{\Phi}{A} \cdot \frac{R_{av}}{1 - R_{av}}, \quad (1.3)$$

Даним методом розраховувати рівень освітленості можливо, коли відомі відбивні властивості поверхні та кількість світлового потоку, що входить в простір.

1.3. Поточковий метод

Цей метод використовується для прямого освітлення, і для його використання повинні бути відомі три таких фактори: інтенсивність джерела світла (в канделах), відстань від джерела світла та орієнтацію поверхні (вертикальна чи горизонтальна). Заснований на законі оберненого квадрата та закону Ламберта [3].

Відповідно, до закону Ламберта яскравість ідеальної розсіюючої поверхні однакова у всіх напрямках та прямо пропорційна косинусу кута між напрямком падіння світла та нормаллю до поверхні.

Закон оберненого квадрата формулюється наступним чином: інтенсивність світла, що виходить з точкового джерела, зворотно пропорційна до квадрата відстані до цього джерела. Тобто, якщо об'єкт зміщений на відстань у 2 рази далі від джерела, він отримує лише чверть тієї інтенсивності, яку він отримував спочатку. Це проілюстровано на рис. 2.

Такий закон може бути використаний тільки на точкових джерелах світла. Тобто для ламп денного світла такий закон не спрацює, оскільки вони не є точковими джерелами.

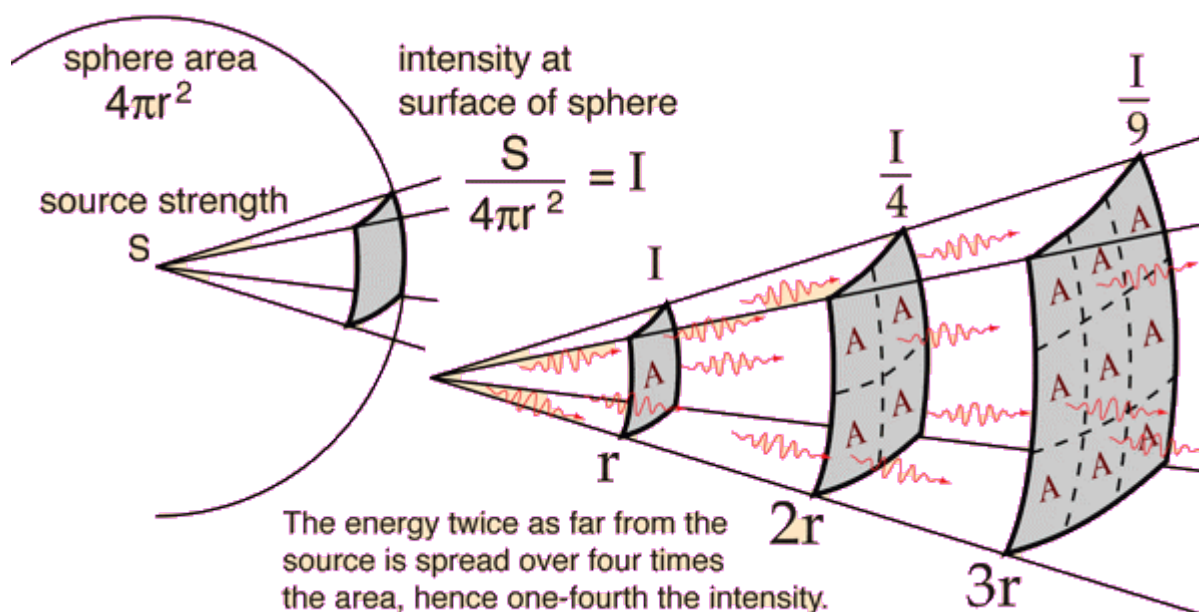


Рис. 2. Закон зворотнього квадрата для обчислення рівня освітлення

Освітленість, в даному випадку, обчислюється за законом косинусів:

$$E = \frac{I}{D^2} \cdot \cos \alpha , \quad (1.4)$$

де α – кут між напрямом падіння світла та нормаллю до поверхні;

I – інтенсивність падаючого світла;

D – відстань до джерела світла.

У застосуванні даного методу існують наступні обмеження [3]:

1. Максимальний фізичний розмір поверхні має бути не більше за

$\frac{1}{5}$ висоти монтажу над точкою оцінки.

2. Не може бути використаний для поверхонь з необмеженою довжиною.

1.4. Узагальнення та порівняльний аналіз існуючих методів

Нижче наведено порівняльну таблицю (табл. 1) методів обчислення освітленості, що розглядалися у даному розділі.

Таблиця 1

Порівняльний аналіз існуючих методів

Назва	Тип освітлення	Додаткові розрахунки	Основа методу	Обмеження
Поточковий метод	Прямий	Інтенсивність джерела світла (в канделях), відстань від джерела світла, орієнтація поверхні (вертикальна чи горизонтальна)	Закон зворотніх квадратів, закон косинусів	Максимальний фізичний розмір поверхні має бути не більше ніж $\frac{1}{5}$ висоти монтажу над точкою оцінки. Не може бути використаний для поверхонь з необмеженою довжиною

Назва	Тип освітлення	Додаткові розрахунки	Основа методу	Обмеження
Метод люмену	Прямий, непрямий	Коефіцієнт утилізації, параметр втрати світла	—	Необхідно додатково визначати коефіцієнт утилізації та параметр втрати світла
Метод Сампнера	Прямий, непрямий	Відбивні властивості поверхні, кількість світлового потоку, що входить в простір	Все світло, яке входить в простір, має або покинути його, або бути поглинутим однією з поверхонь простору	Мають бути відомі відбивні властивості поверхні та кількість світлового потоку, що увійшло в простір

Проаналізувавши табл. 1, можна зробити висновок, що, не зважаючи на необхідність у додаткових обчисленнях параметрів, метод люмену найбільш підходить для обчислень рівня освітлення приміщення.

1.5. Висновки до розділу 1

В даному розділі було розглянуто питання ергономічності приміщення та його важливість при проектуванні приміщення. Дане питання розділяється на три основні підпитання, що є найбільш важливими для ергономіки приміщення. Було обрано питання оптимального розміщення джерел світла.

Для розв'язання даного питання були розглянуті три основних методи, що дозволяють обчислити рівень освітленості приміщення, а саме: метод люмену, метод Сампнера та поточковий метод. Було зазначено, що світло буває прямим та непрямим, а його відбиття – дифузним та дзеркальним.

На підставі наведених у порівняльній таблиці даних було визначено, що для вирішення задачі оптимізації розташування джерел освітлення приміщення найбільш придатним для виконання обчислень є метод люмену.

2. РОЗРОБЛЕННЯ СПОСОБУ РОЗМІЩЕННЯ ДЖЕРЕЛ ОСВІТЛЕННЯ

2.1. Постановка науково-інноваційної задачі

Задача полягає у розробленні інноваційного програмного продукту, що реалізує спосіб оптимального розміщення джерел освітлення та забезпечує:

- проведення перевірки рівня освітлення в приміщенні;
- проведення оптимізації розміщення джерел освітлення з метою підвищення ергономічності приміщення в сенсі світла;
- повідомлення про порушення норми рівня освітлення, що описані в стандартах, наведених нижче.

Розраховується глобальний рівень освітлення – тобто середній рівень освітлення в приміщенні.

2.2. Формулювання гіпотез

Виходячи з розглянутих методів та поставленої мети, можна сформулювати наступні гіпотези:

- рівень освітлення можна обчислити, використовуючи метод люмену;
- обчислення можна проводити в реальному часі;
- за допомогою розрахунку рівня освітлення в приміщенні можна оптимально розмістити джерела світла, підвищивши таким чином ергономічність самого приміщення.

2.3. Аналіз стандартів освітлення приміщень

Для України існують державні будівельні норми ДБН В.2.5-28:2018 “Природне та штучне освітлення” [5], затверджені наказом Міністерства регіонального розвитку, будівництва та житлово-комунального господарства України від 03.10.2018 та чинні від 01.03.2019. В даному документі наведені вимоги до освітлення приміщень виробничих

підприємств, житлових, цивільних та адміністративно-побутових споруд, в залежності від характеристики зорової роботи, розряду, підрозряду зорової роботи тощо. Наприклад, для приміщень виробничих підприємств із характеристикою зорової роботи найвищої точності з різною підрозрядністю освітленість при системі загального освітлення (тобто таке, за якого світильники розміщуються рівномірно у верхній частині приміщення) становить в діапазоні від 300 до 1200 лк.

Нижче наведені спрощені таблиці вимог до освітлення для різних приміщень, в залежності від характеристики зорової роботи та підрозряду зорової роботи, для кожного типу приміщень. Дані критерії є спільними як для приміщень виробничих підприємств, так й для приміщень житлових, цивільних та адміністративно-побутових споруд.

Таблиця 2

Вимоги до освітлення приміщень виробничих підприємств

Характеристика зорової роботи	Підрозряд зорової роботи	Освітленість, лк
Найвищої точності	а	—
	б	1000-1200
	в	600-750
	г	300-400
Дуже високої точності	а	—
	б	600-750
	в	400-500
	г	200-300
Високої точності	а	400-500
	б	200-300
	в	200-300
	г	200
Середньої точності	а	300
	б	200
	в	200
	г	200

Продовження табл. 2

Характеристика зорової роботи	Підрозряд зорової роботи	Освітленість, лк
Малої точності	а	300
	б	200
	в	200
	г	200
Груба (дуже малої точності)	—	200

Таблиця 3

Вимоги до освітлення приміщень житлових, цивільних та адміністративно-побутових споруд

Характеристика зорової роботи	Підрозряд зорової роботи	Освітленість, лк
Розрізнення об'єктів при фіксованій та нефіксованій лінії зору:		
— дуже високої точності	1	500
	2	400
— високої точності	1	300
	2	200
— середньої точності	1	150
	2	100
Огляд оточуючого простору при дуже короткочасному епізодичному розрізненні об'єктів:		
— при високій насиченості приміщень світлом	—	300
— при нормальній насиченості приміщень світлом	—	200
— при низькій насиченості приміщень світлом	—	150
Загальне орієнтування в просторі інтер'єру:		
— при великому скупченні людей	1	75
— при малому скупченні людей	2	50
Загальне орієнтування в зонах пересування:		
— при великому скупченні людей	1	30

Характеристика зорової роботи	Підрозряд зорової роботи	Освітленість, лк
– при малому скупченні людей	2	20

В загальних положеннях цього документу зазначено, що середня освітленість робочих місць, де постійно перебувають люди, повинна бути не менше ніж 200 лк. Також такі приміщення повинні мати природне освітлення, що зазначено у розділі “Природне освітлення”. Винятком є ті приміщення, які визначені відповідними державними будівельними нормами та стандартами, а також ті, розміщення яких дозволено в підвальних поверхах будівель. У приміщеннях може застосовуватися бокове, верхнє і комбіноване (верхнє та бокове), транспортоване та акумульоване природне освітлення. В даній дисертації використовується лише бокове освітлення. Мінімальне значення КПО у приміщеннях житлових та громадських будівель при боковому освітленні з однієї сторони забезпечується в розрахунковій точці робочої поверхні, найбільш віддаленій від вікон. Вона лежить на перетині площини характерного розрізу на відстані 1 м від стіни, протилежної вікнам, та робочої поверхні.

Робочою поверхнею є:

- у житлових приміщеннях житлових будинків та інших приміщеннях невиробничого характеру – підлога;
- у навчальних і навчально-виробничих приміщеннях шкіл, професійно-технічних і вищих навчальних закладів, у кабінетах лікарів, у перев'язочних – умовна робоча поверхня, що розташована на висоті 0,8 м над підлогою;
- в інших приміщеннях різного призначення – згідно з додатком Д в даному стандарті.

При односторонньому боковому освітленні у виробничих приміщеннях глибиною до 6 м нормується мінімальне значення КПО, яке забезпечується у розрахунковій точці умовної робочої поверхні, що

знаходиться на перетині цієї поверхні та вертикальної площини характерного розрізу приміщення на відстані 1 м від стіни, протилежної вікнам, або в найбільш віддаленій від вікон точці робочої поверхні, в якій триває виробничий процес. У великогабаритних виробничих приміщеннях (більше ніж 6 м глибини) мінімальне значення КПО при боковому освітленні нормується в точці на умовній робочій поверхні, віддаленій від світлових прорізів:

- на 1,5 висоти від підлоги до верху світлових прорізів для зорової роботи I-IV розрядів;
- на 2 висоти від підлоги до верху світлових прорізів для зорової роботи V-VII розрядів;
- на 3 висоти від підлоги до верху світлових прорізів для зорової роботи VIII розряду.

У таких приміщеннях доцільно застосовувати спеціальні світловідбивні екрани та жалюзі на вікнах, що перерозподіляють світловий потік в глибину приміщення.

Нормовані значення КПО треба визначати залежно від призначення приміщень за таблицями 5.1, 5.2 та додатками Д і Ж, що зазначені в стандарті.

При застосуванні освітлення крізь вікна, що розташовані у кількох стінах (крім виробничих приміщень глибиною 6 м і більше), мінімальне значення КПО повинно бути забезпечено у найменш освітленій точці робочої поверхні по характерному розрізу приміщення. При боковому двосторонньому освітленні таких приміщень та однакових вікнах з обох сторін дозволяється за розрахункову точку приймати таку, що розташована в центрі приміщення на перетині вертикальної площини характерного розрізу і робочої поверхні.

Для приміщень, де виконується зорова робота на певних негоризонтальних поверхнях (наприклад, класна дошка в аудиторіях), крім горизонтальної робочої поверхні, природне освітлення нормується і на цих

поверхнях. Положення додаткових розрахункових точок визначається відповідно до додатка Д, зазначеному у стандарті.

Розрахунок КПО виконується з урахуванням середньозважених коефіцієнтів відбивання світла внутрішніми поверхнями приміщень та фасадів протилежних будівель та споруд, але без урахування меблів, устаткування, обладнання, озеленення та інших затінюючих предметів, а також при 100% використанні світлопрозорих заповнень у світлопрорізах.

Розрахункові значення КПО слід заокруглювати до сотих часток. Методика розрахунку КПО визначається відповідними стандартами в залежності від виду природного освітлення. Дозволяється зниження розрахункового значення КПО від нормованого не більше ніж на 10%. Розрахункові значення середньозваженого коефіцієнта світловідбивання внутрішніх поверхонь приміщення слід приймати на підставі прийнятої в проекті архітектурної обробки поверхонь, але не більше 0,5 – у приміщеннях цивільних будівель та кухнях житлових будинків, а також у виробничих приміщеннях зі світлою характеристикою фону. У житлових кімнатах житлових будинків та у виробничих приміщеннях із середньою характеристикою фону – не більше 0,4; не більше 0,30 – у виробничих приміщеннях з темною характеристикою фону.

Для освітлення природним світлом приміщень, що не мають зовнішніх огорожень, а також зон приміщень, віддалених від світлопрорізів, рекомендується застосовувати світловоди. У приміщеннях великої глибини, розташованих на останньому поверсі, де природне бокове освітлення не дозволяє забезпечити нормоване значення КПО або бічне освітлення неможливо влаштувати, а велика висота покриття не дозволяє використати зенітні ліхтарі рекомендується влаштовувати світлові шахти.

Використання природного освітлення у нічний час забезпечується системами, що акумулюють природне освітлення, та їх розрахунок проводиться за нормами штучного освітлення.

Окрім вимог, в цьому стандарті приведені нормовані показники освітлення різних типів приміщень, в залежності від площини нормування освітленості та її висоти над рівнем підлоги, та від розряду та підрозряду зорової роботи. Нижче наведена таблиця цих показників для житлових приміщень та приміщень цивільних будівель, такі як кабінети, робочі кімнати, офіси, приміщення обслуговуючого персоналу, лабораторії тощо.

Таблиця 4

Нормовані показники освітлення основних приміщень цивільних будівель

Приміщення	Орієнтація площини та її висота над рівнем підлоги, м	Розряд та підрозряд зорової роботи	Освітленість, лк
Кабінети, робочі кімнати і офіси	Г – 0,8	Б-1	300
Проектні зали і кімнати	Г – 0,8	А-1	500
Книгосховища й архіви	В – 1,0 на стелажах	–	–
Приміщення для роботи з дисплеями і відеотерміналами	В – 1,2 на екрані дисплея	Б-2	200
	Г – 0,8	А-2	400
Читальні зали	Г – 0,8	А-2	400
Класні кімнати, аудиторії, навчальні кабінети, лабораторії закладів середньої освіти	В – 1,5 на середині дошки	А-1	500
	Г – 0,8 на робочих столах і партах	А-2	400
Аудиторії, навчальні кабінети, лабораторії у вищих навчальних закладах	В – 1,5 на середині дошки	А-2	400
	Г – 0,8 на робочих столах і партах	А-2	400
Кабінети інформатики і обчислювальної техніки	В – 1,0 на екрані дисплея	Б-2	200
	Г – 0,8 на робочих столах і партах	А-2	400

Нормовані показники освітлення основних приміщень житлових будинків

Приміщення	Орієнтація площини та її висота над рівнем підлоги, м	Розряд та підрозряд зорової роботи	Освітленість, лк
Житлові кімнати, вітальні, спальні, житлові кімнати гуртожитків	Г – підлога	В-1	150
Кухні, кухні-їдальні	Г – 0,8	В-1	150
Дитячі	Г – підлога	Б-2	200
Кабінети, бібліотеки	Г – 0,8	Б-1	300
Ванні кімнати, вбиральні, санвузли, душові, внутрішньоквартирні коридори	Г – підлога	Ж-2	50
Гардеробні	Г – підлога	Ж-1	75
Басейни	Г – поверхня води	В-2	100
Тренажерний зал	Г – підлога	В-1	150
Більярдна	Г – 0,8	Б-1	300

Слід зазначити, що наведені показники визначені лише для штучного освітлення.

2.4. Аналіз методів оптимізації

В математиці задача оптимізації полягає у знаходженні екстремуму заданої функції – її мінімуму або максимуму. Оптимізацію можна поділити на параметричну та структурну. Параметрична оптимізація пов'язана з розрахунком оптимальних значень параметрів при заданій структурі об'єкта. Структурна оптимізація пов'язана із задачею вибору оптимальної структури.

Методи оптимізації класифікують відповідно до задач оптимізації:

- локальні методи: сходяться до локального екстремуму цільової функції;
- глобальні методи: мають справу з багатоекстремальними функціями.

За вимогами до гладкості й наявності в цільовій функції частинних похідних методи можна поділити на:

- прямі методи, що вимагають тільки обчислень в точках наближення;
- методи першого порядку: вимагають обчислень перших частинних похідних функції;
- методи другого порядку: вимагають обчислень других частинних похідних функції.

Також вони поділяються на:

- аналітичні методи;
- чисельні методи;
- графічні методи.

За природою множини X , а також за видом цільової функції та її обмежень методи поділяються на такі класи:

- якщо множина X скінченна або зліченна, задача розв'язується методами дискретного програмування;
- якщо X є підмножиною множини цілих чисел – методи цілочислового програмування;
- якщо обмеження або цільова функція містять нелінійні функції, а множина X є підмножиною скінченновимірною векторного простору – методи нелінійного програмування;
- якщо усі обмеження та цільова функція є лінійними – методи лінійного програмування.

Крім цих методів, існують також методи параметричного, динамічного та стохастичного програмувань. Також вони розбиваються на три великі групи: детерміновані, випадкові (стохастичні) та комбіновані.

Основою для рішення задачі оптимізації є математичні моделі. Математичною моделлю називається сукупність математичних співвідношень, що описують суть завдання.

Математичну модель задачі лінійного програмування можна сформулювати таким чином: максимізувати/мінімізувати лінійну функцію, що має лінійні обмеження. Тобто:

$$\begin{aligned} \sum_{j=1}^m c_j x_j \rightarrow \max \vee \min, \\ \sum_{j=1}^n a_{ij} x_j \leq b_j, \quad i = \overline{1, m}, \end{aligned} \quad (2.1)$$

де $x_j \geq 0$.

Математична модель задачі цілочислового програмування можна сформулювати наступним чином:

$$\begin{aligned} \sum_{j=1}^m c_j x_j \rightarrow \max \vee \min, \\ \sum_{j=1}^n a_{ij} x_j \leq b_j, \quad i = \overline{1, m}, \end{aligned} \quad (2.2)$$

де $x_j \geq 0$, x_j – цілі числа.

Як видно, дана модель майже не відрізняється від задачі лінійного програмування, за винятком того, що на розв'язок лінійного програмування накладається додаткове обмеження – визначення лише цілих чисел в значеннях змінних.

Аналогічно з математичною моделлю для задачі нелінійного програмування. Вона схожа з моделлю задачі лінійного програмування, однак для даної задачі необхідно умова нелінійності цільової функції або ж обмежень.

2.5. Формулювання способу оптимального розміщення джерел освітлення

Для вирішення задачі оптимального розміщення джерел світла у приміщенні можна було б застосувати будь-який із методів, що були

наведені в підрозділі 2.4. Але умови задачі дозволяють уникнути застосування методів оптимізації з побудовою відповідних математичних моделей, формування критеріїв, обмежень тощо.

Оскільки для розрахунків, що стосуються рівня освітлення приміщення обрано метод люмену (на підставі виконання аналізу методів розрахунку рівня освітлення в приміщенні) сформулюємо спосіб оптимального розміщення світильників в приміщенні для вирішення задачі програмними засобами.

Для оптимізації розташування джерел штучного освітлення нехтуємо природне освітлення і навпаки.

Суть методу полягає в тому, що площа приміщення ділиться на квадрати (рис. 3), розмір яких залежить від потужності джерел штучного світла (можуть застосовуватися світильники накаливання потужністю від 40 до 100 Вт та світильників денного світла потужністю 18 та 36 Вт). Цей квадрат є поверхнею яку освітлює світильник і рівень освітленості цієї поверхні повністю відповідає нормам будівельного стандарту ДБН В.2.5-28:2018 “Природне та штучне освітлення”.

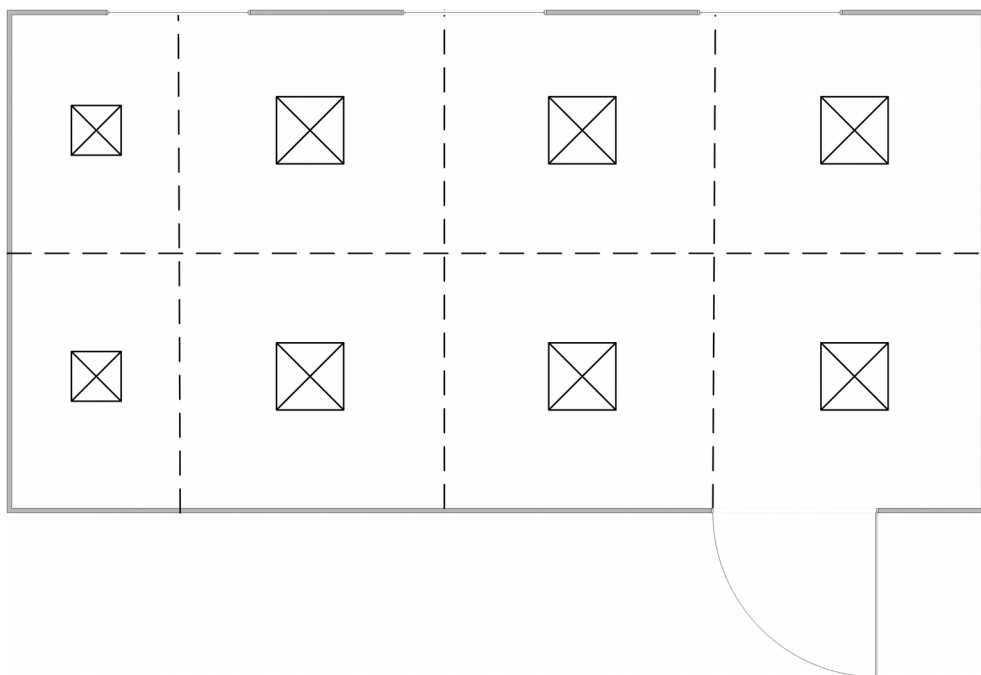


Рис. 3. Приклад приміщення без внутрішніх перегородок та розташування світильників

Якщо приміщення має внутрішні перегородки, то перед виконанням розрахунків площа приміщення розділяється на частини, відповідно до кількості та розташування перегородок. Для кожної з частин приміщення проводимо оптимізацію, яка полягає у розділенні площі приміщення або окремої його частини на квадрати, площа яких відповідає потужності обраного штучного джерела світла. Кількість квадратів, їх розмір та розміщення в приміщенні або окремій частині приміщення буде дорівнювати кількості світильників, а площа квадратів визначить їх потужність. Приклад із внутрішньою перегородкою (рис. 4) показує даний принцип.

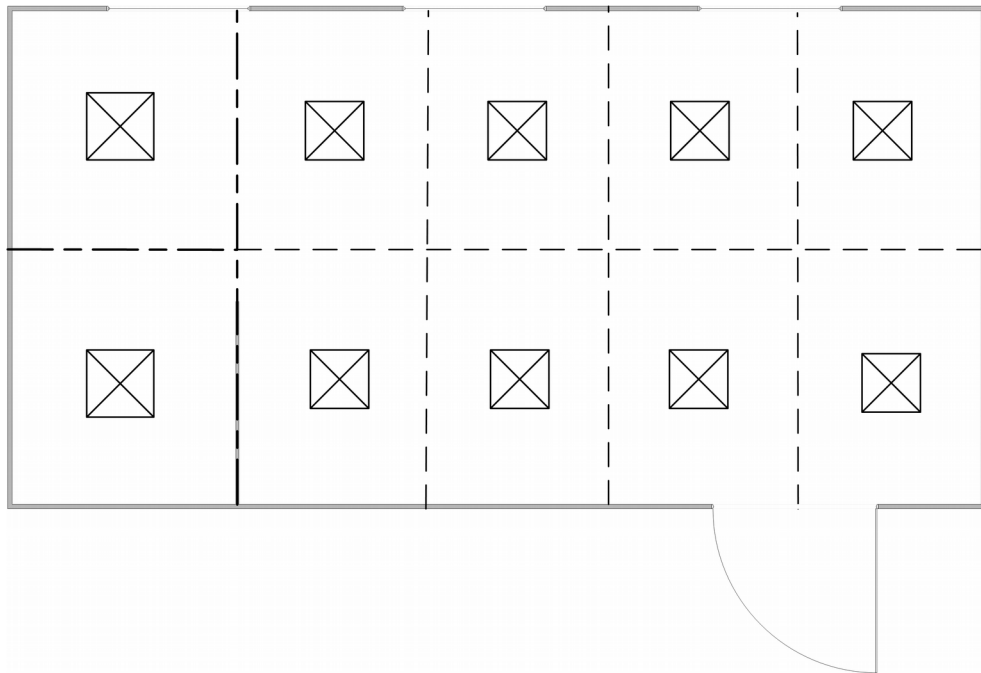


Рис. 4. Приклад приміщення з внутрішньою перегородкою та розташування світильників

Оптимізація джерел природного освітлення проводиться наступним чином. Виконується розрахунок освітлення приміщень відповідно до заданих розмірів вікон та заданої їхньої кількості.

Оптимальна кількість та розмір вікон бокового освітлення розраховується на основі обчислення рівня освітлення приміщення за рівнем КПО, яке вираховується за формулою:

$$E = \frac{S_{\text{в}} \cdot t_0 \cdot t_1 \cdot r_1 \cdot 100}{m \cdot K_{\text{буд}} \cdot K_{\text{з}} \cdot n_{\text{в}} \cdot S_{\text{п}}}, \quad (2.3)$$

де $S_{\text{в}}$ – площа вікон;

t_0 – коефіцієнт світлопропускання скла (приймається рівним 0,8);

t_1 – коефіцієнт світлопропускання рами (приймається рівним 0,7);

r_1 – коефіцієнт відбиття;

m – світловий клімат (приймається рівним 0,85);

$K_{\text{буд}}$ – коефіцієнт затінення вікон будівлями (приймається рівним 1,2);

$K_{\text{з}}$ – коефіцієнт запасу (приймається рівним 2)

$n_{\text{в}}$ – світлова характеристика вікон;

$S_{\text{п}}$ – площа підлоги;

E – значення КПО, %.

За тим вікна розташовуються рівномірно по зовнішніх стінах приміщення.

2.6. Висновки до розділу 2

У даному розділі поставлено науково-інноваційну задачу, яка полягає у розробленні програмного забезпечення, що реалізує метод оптимального розміщення джерел світла, проаналізовано стандарт ДБН В.2.5-28:2018 “Природне та штучне освітлення” та виділено основні вимоги до освітлення для приміщень різного типу, а також норми освітлення.

Окрім цього, були проаналізовані методи оптимізації. Вони бувають локальними та глобальними, прямими, першого та другого порядку, аналітичними, чисельними та графічними. Також методи діляться за типом цільової функції на методи лінійного програмування, методи дискретного програмування, цілочислового програмування, нелінійного програмування, параметричного, стохастичного та динамічного програмувань.

На основі аналітичної інформації, що наведена в розділі 2, були розроблені способи оптимального розташування джерел штучного та природного освітлення.

3. РОЗРОБЛЕННЯ ІННОВАЦІЙНОГО ПРОГРАМНОГО ПРОДУКТУ

3.1. Вимоги до інноваційного програмного продукту

Основною вимогою до інноваційного програмного продукту – забезпечити можливість змодельовати приміщення з оптимально розташованими джерелами світла у ньому, а також надати можливість інтеграції з наявними 3D-редакторами моделей та підтримку основних ОС, такі як Windows, Linux та MacOS.

Також програмний продукт повинен надавати користувачеві такі можливості:

- вводити вхідні дані, які застосовуються для розрахунку в графічному інтерфейсі, а саме:
 - розміри приміщення: ширина, глибина, висота від підлоги до світильника в метрах;
 - тип робіт за точністю (висока, середня, низька, груба);
 - розмір перегородок (довжина, ширина в метрах), їх кількість;
 - вікна, кількість та розмір: ширина, висота в метрах (вікна можуть бути різного розміру), кількість вікон задається для кожного їх розміру;
 - на якій стіні розташоване вікно (фронтальна, ліва, права);
 - висота до нижньої кромки вікна та висота до верхньої кромки вікна в метрах;
 - тип світильників за потужністю (40 Вт, 60 Вт, 100 Вт, T8: 36 Вт, T5: 18 Вт) та їх кількість;
 - розташування світильників (зліва, в глибину, в метрах);
 - розташування перегородок (початкова точка на стіні, кінцева – кінець довжини);
 - розташування вікон (ширина вікон, відстань від початку стіни);
- перевіряти рівень освітлення в приміщенні;
- проводити оптимізацію розміщення джерел світла;

- отримувати повідомлення про відповідність до норм освітлення;
- отримувати повідомлення про можливість та необхідності оптимізації.

З нефункціональних вимог програмний продукт повинен забезпечувати:

- графічний інтерфейс для взаємодії користувача з програмним продуктом;
- інтеграцію з 3D-редактором;
- виконання з середовища 3D-редактора, який, у свою чергу, підтримує основні ОС: Windows, Linux, MacOS.

Узагальнення функціональних вимог до продукту подано у вигляді варіантів використання на рис. 5.

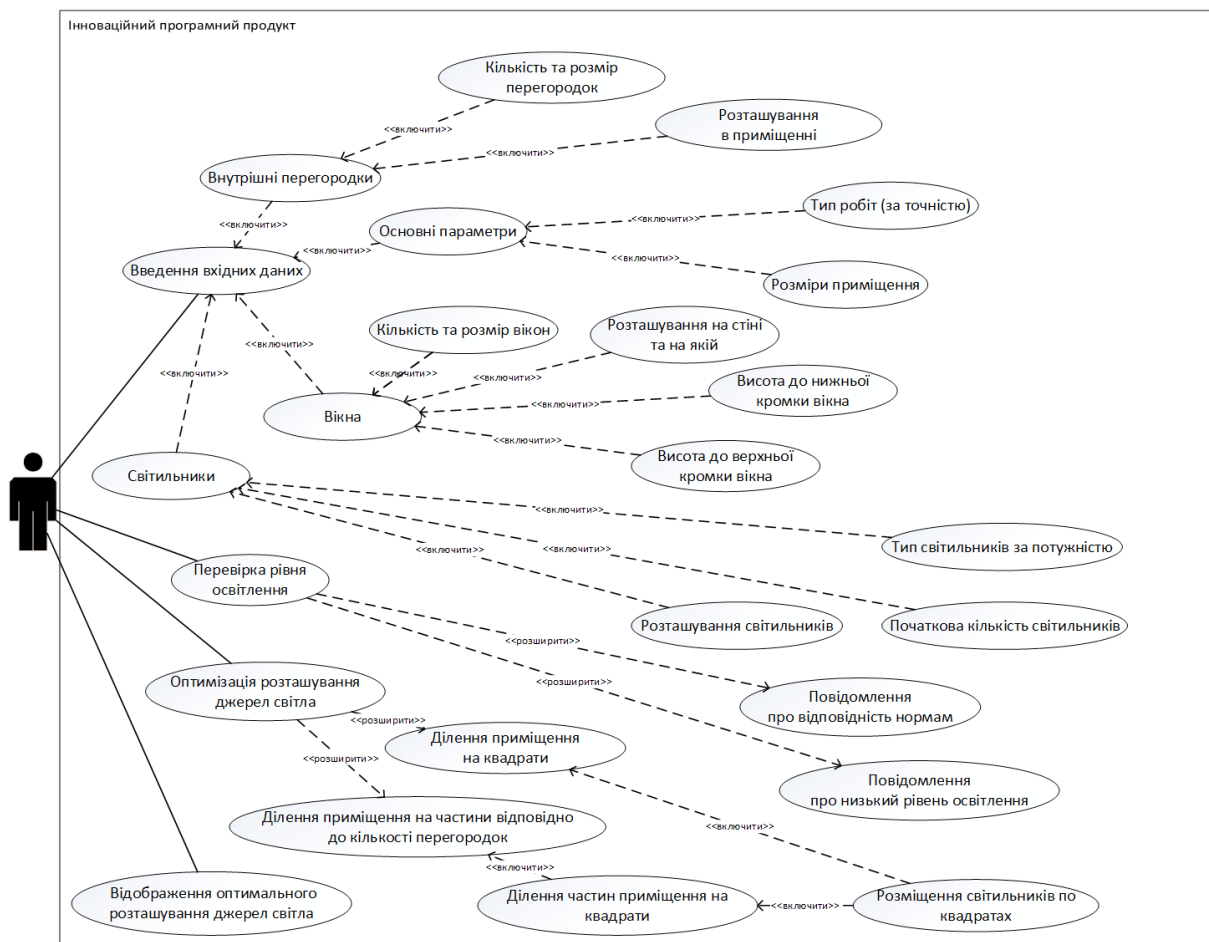


Рис. 5. Варіанти використання програмного продукту

3.2. Вибір засобів розроблення

Оскільки програмний продукт є доповненням до 3D-редактора, необхідно обрати найбільш відповідний до вимог. Даними редакторами є безкоштовний з відкритим вихідним кодом Blender, редактори компанії Autodesk (Maya, 3ds Max, Revit, AutoCAD) та SketchUp. Кожний з них надає можливість розробнику інтегрувати власне програмне забезпечення та мають повну документацію.

Редактори компанії Autodesk є достатньо потужними для реалізації задач 3D-моделювання. Кожний з цих продуктів надає можливість розроблювати та інтегрувати власні доповнення.

Autodesk Maya надає потужні програмні інструменти для 3D-моделювання, а саме:

- Bifrost – середовище візуального програмування, дозволяє швидко і просто створювати процедурні ефекти, гідні блокбастерів:
 - алгоритми для моделювання і візуалізації фотореалістичних рідин (Bifrost Fluids);
 - глибоке адаптивне моделювання потоків рідини;
 - система моделювання морської поверхні;
- готові до використання анімаційні криві – дозволяють створювати приголомшливі ефекти, наприклад снігові та пилові бурі;
- інтерактивне укладання волосся – реалістичні ефекти згущення волосся і вовни, що додає їм більш природного вигляду.
- створення реалістичною морської поверхні за допомогою хвиль, мерехтіння і завихрень.
- технологія Maya nCloth – створення реалістичних деформуються матеріалів;
- технологія Bullet Physics – створення реалістичних моделей твердих і аморфних тіл;
- відтворення кешованого вмісту – завдяки більш швидкому відтворенню анімації в видовому екрані Viewport 2.0 можна

виконувати її перевірку безпосередньо на видовому екрані, що зводить до мінімуму необхідність створення візуалізацій за допомогою Playblast;

- геодезичне зрощування вокселів – створення високоякісних, готових до використання пов'язаних персонажів за менший час;
- базові інструменти анімації – інструментарій для анімації по ключовим кадрам, а також процедурної і програмованої анімації;
- редактор часу – недеструктивний нелінійний редактор, який працює з кліпами, для підготовки висококласних анімацій;
- робочий процес для розробки форм – швидке і зручне налаштування персонажів і анімації;
- створення анімації – покращені можливості швидкого створення сцен;
- паралельне обчислення оснащення – вбудований засіб профілювання продуктивності забезпечує більш швидке відтворення та маніпулювання;
- захоплення рухів – фільтри Butterworth і Key Reducer, які забезпечують додаткову точність анімаційних кривих;
- робоче середовище – можливості Arnold дозволяють максимально наблизити анімацію при попередньому перегляді в Viewport 2.0 до підсумкової візуалізації Arnold, що суттєво заощаджує час і дозволяє підвищити ефективність творчого процесу;
- інтеграція Arnold і Maya – використання Arnold RenderView для перегляду в сценах зміни освітлення, матеріалів і руху камери, в режимі реального часу;
- Hypershade – спрощене тонування складних сцен;
- поліпшене керування кольором та підвищена узгодженість логічних операцій;

- потужний редактор UV-текстур – інтерфейс редактора UV-текстур включає комплект вдосконалених інструментів і функцій для роботи з UV-текстурами;
- інструментарій для створення скульптур – забезпечує зручне створення скульптурних моделей і зміну їх форм;
- моделювання полігонів – повнофункціональна бібліотека, що дозволяє виконувати логічні операції в геометрії полігонів швидше і більш злагоджено;
- підтримка OpenSubdiv – збільшена продуктивність за допомогою інтерактивних робочих процесів;
- вузли MASH – інструментарій MASH з доданими новими вузлами (кривої, сигналом, натовпом, конструктором і багатьом ін.) і вдосконаленими існуючими;
- брендинг, розробка літаючих логотипів, послідовностей заголовків і інших проектів, для яких потрібне створення тексту;
- робочий процес використання векторної графіки – імпорт, копіювання і вставка файлів SVG в Maya;
- інструментарій для створення графіки руху – швидке створення складних процедурних ефектів і анімації за допомогою об'єктів-посилань;
- гнучкий робочий процес – спрощене створення, налагодження та інтеграція Maya до виробничих процесів підприємства;
- створення сценаріїв і API-інтерфейс – створення сценаріїв Maya і модулів на мовах Maya Embedded Language (MEL) і Python.
- інструменти управління даними та сценами – управління великими наборами даних і складними сценами за допомогою спеціалізованих інструментів і робочих процесів;
- інструменти збірки сцени для інтелектуальних даних – спрощене створення великих складних графічних середовищ і управління виробничими компонентами як окремих елементів.

Можливостей роботи з графікою у 3D-редактора Autodesk Maya набагато більше, ніж це необхідно для застосування при вирішенні задач оптимізації розміщення джерел світла в приміщенні, але, за необхідності він цілком може бути використаний для рішення цієї задачі. Він цілком підходить для автоматизації дій, які повторюються, та для розширення основної функціональності редактора. Це включає в собі власні налаштування для 3D-моделювання, анімації, візуальних ефектів, рендерингу тощо [6]. Додати своє доповнення можна за допомогою мови програмування або MEL (Maya Embedded Language), або Python [7].

Графічний 3D-редактор Autodesk 3ds Max призначений для 3D-моделювання та візуалізації. Він також дозволяє працювати з візуалізацією проектів, іграми та анімацією. Розглянемо його основні функції:

- робота зі сплайнами – дозволяє створювати та анімувати геометрію простими способами за допомогою нових і розширених інструментів;
- підтримка Open Shading Language – для створення карти OSL в редакторі матеріалів від простих математичних вузлів до повних процедурних текстур;
- Blended Box Map – дозволяє змінювати видимі стики, спрощуючи процес змішування спроектованих карт текстур;
- редагування фасок – дозволяє створювати кращі в своєму класі деталі процедурного моделювання, щоб легко виконувати деякі найскладніші завдання;
- модифікатор волоссяного і хутряного покриву – за допомогою інструментів вибору і стилізації спрощує керування волоссяним і хутряним покривом безпосередньо в видовому екрані, в тому числі стрижки і розчісування;
- Data Channel Modifier – при необхідності застосування процедурної зміни дозволяє отримати інформацію про вершини, ребра і грані моделі;

- формування об'єктів логічних операцій – процес, що призначений для створення параметричних логічних операцій за двома або більше сплайнами в знайомому інтерфейсі логічних 3D-об'єктів;
- моделювання мереж і поверхонь – допоможе в оптимізації створення параметричних і органічних об'єктів за допомогою функцій моделювання на основі полігонів, поверхонь поділу і сплайнів;
- 3ds Max Fluids – призначений для відображення реалістичної поведінки рідин безпосередньо в 3ds Max;
- траєкторії руху – дозволяє керувати анімацією безпосередньо в видовому екрані з отриманням зворотного зв'язку безпосередньо під час налаштування сцен;
- інструменти анімації та оснащення персонажів – дозволяють створити процедурну анімацію і виконувати оснащення персонажів за допомогою цього набору, а також набору інструментів для анімації двоногих персонажів і масовки;
- інструменти для загальної анімації – засоби процедурної анімації та анімації по ключовим кадрам, які дозволяють здійснювати перегляд і редагування анімаційних траєкторій безпосередньо на видовому екрані;
- контролери анімації в розширеному середовищі створення графів – розробляються за допомогою інструментів нового покоління, які можна створювати, змінювати, поміщати в пакети і якими можна ділитися з іншими користувачами;
- ефекти руху частинок – дозволяються створювати реалістичні ефекти частинок, наприклад воду, вогонь, бризки і сніг;
- простий імпорт даних моделювання – дозволяє створювати анімацію даних моделювання в форматах CFD, CSV і OpenVDB;

- інструменти скіннінга Geodesic Voxel і Heatmap – призначені для швидкого і не обтяжливого створення якісного розрахунку ваги оболонки;
- покращена якість видового екрана – дозволяє відображати процедурні карти, створені за допомогою OSL, зберігаючи вихідну якість, за допомогою видового екрана Nitrous;
- Arnold для 3ds Max – плагін MAXtoA інтегрований в 3ds Max, що забезпечує доступ до останніх функцій Arnold;
- фізична камера – дозволяє моделювати налаштування фізичної камери, наприклад витримку, апертуру, глибину різкості, експозицію та інші параметри;
- видовий екран ActiveShade – призначений для керування компонентами сцени прямо в підсумковому модулі візуалізації ActiveShade;
- засіб візуалізації Autodesk Raytracer (ART) – дозволяє створити точні зображення архітектурних об'єктів;
- компонування сцен у віртуальній реальності – призначений для редагування сцен безпосередньо в віртуальній реальності 3ds Max Interactive і дає можливість спостерігати за оновленням проекту 3ds Max в реальному часі;
- 3ds Max Batch – можливості автоматизації 3ds Max, шляхом забезпечення роботи з програмою з командного рядка;
- взаємодія робочих інструментів – більш тісна реалізація взаємодії деяких робочих інструментів з розширеним і вдосконаленим інструментарієм Python / .NET;
- підтримка формату Alembic – засіб імпортування матеріалів Alembic, який дозволяє виконати перевірку файлів перед виконанням імпорту;

- сучасний інтерфейс і робочі простору – створюйте користувальницькі робочі простору з допомогою сучасного чуйного інтерфейсу з підтримкою високої роздільної здатності;
- конвертер сцен – дозволяє безперешкодно переходити від одного модуля візуалізації до іншого;
- оновлення робочого процесу Civil View – транспортні засоби, створені в Civil View, тепер підтримують матеріали 3ds Max, що скорочує перетворення і спрощує візуалізацію.

Autodesk 3ds Max має достатню кількість функцій для вирішення задачі оптимізації розташування джерел освітлення приміщення, а також надає програмний інтерфейс для розробки, який використовується для автоматизації дій та розширення функціональності редактора. Розроблювати власні доповнення необхідно за допомогою мов програмування C++, C# або Python [8].

Revit – програмне забезпечення для інформаційного моделювання будівель. Він призначений для створення інтелектуальних моделей, необхідних для планування, проектування, будівництва та експлуатації будівель і об'єктів інфраструктури. Revit підтримує процес міжгалузевого проектування і забезпечує його середовищем для спільної роботи. Основні можливості його такі:

- параметричні компоненти – створюють відкриту графічну систему для проектних рішень і моделювання форм, а також є основою для всіх будівельних компонентів, розроблених за допомогою програмного забезпечення Revit;
- спільна робота над проектом – забезпечення учасникам проекту, які працюють в різних сферах, спільного використання програмного забезпечення і проектів з можливістю збереження своїх даних в одному централізованому загальному файлі;

- специфікації в Revit – табличне представлення інформації про модель, шляхом імпорту інформації з властивостей елементів проекту;
- Revit дозволяє імпортувати, експортувати і пов'язувати дані в часто використовуваних форматах;
- взаємодія і IFC – імпорт, експорт і пов'язування даних з IFC4 та іншими часто використовуваними форматами;
- надбудови – розширення функціональності Revit за рахунок доступу до API-інтерфейсів, сторонніх рішень, надбудов і бібліотек компонентів в Autodesk App Store;
- анотація – призначена для більш ефективної демонстрації проектів завдяки підтримці редагування WYSIWYG і можливостям, які дозволяють керувати відображенням тексту;
- Dynamo for Revit – призначений для розгортання і оптимізації робочих процесів BIM за допомогою графічного інтерфейсу програмування з відкритим вихідним кодом, який встановлюється разом з Revit;
- глобальні параметри – реалізація налаштувань, до яких відносяться радіальні і діаметральні розміри і залежності рівності;
- інструменти для концептуального проектування – створення ескізів і моделей довільної форми, а також формотворчих елементів;
- аналіз за допомогою Insight – оптимізація експлуатаційних характеристик будівель за рахунок централізованого доступу до даних про продуктивність і поліпшеним механізмам аналізу;
- архітектурне моделювання – додавання архітектурних елементів в модель будівлі, таких як стіни, двері, вікна, компоненти тощо;
- інструменти роботи з хмарами точок – застосування даних лазерного сканування безпосередньо в процесі інформаційного моделювання будівель для створення виконавчої моделі;

- візуалізація 3D-проекту – дослідження, затвердження і демонстрація проектів із застосуванням більш точної і швидкої візуалізація проекту за допомогою засобу Autodesk Raytracer;
- багатоповерхові сходи – забезпечення можливості швидкого створення і редагування багатоповерхових будівель шляхом прив'язки сходів до рівнів в проекті;
- хмарна візуалізація – забезпечення формування фотореалістичних візуалізацій без прив'язки до комп'ютера компанії та без використання спеціального обладнання;
- проектування і виготовлення несучих конструкцій (фізична і аналітична модель) – створення фізичної моделі для координації та підготовки документації, а також пов'язаної з нею аналітичної моделі для здійснення розрахунку будівельних конструкцій;
- деталізація арматури – створення 3D-моделей арматури для монолітних і збірних залізобетонних конструкцій, забезпечення підготовки та випуску проектної документації з армування з включенням в неї специфікації арматурних стержнів;
- конструювання металоконструкцій – забезпечення процесу моделювання зв'язок з більш високим рівнем деталізації з використанням параметричних сталевих з'єднань, які можна знайти у великій бібліотеці Revit або застосування створених самостійно об'єктів;
- будівельна документація – розробка точної і докладної будівельної документації для проектів, що включають різні матеріали;
- двосторонній зв'язок з аналізом – інтеграція результатів аналізу в процес BIM і робота в ітераційному процесі проектування;
- підключення до процесів виробництва залізобетонних конструкцій – CAM Export забезпечує підключення проектів монолітних і збірних залізобетонних конструкцій до автоматичних процесів виготовлення;

- Dynamo для проектування будівельних конструкцій – Dynamo надає розробникам будівельних конструкцій, проектувальникам і деталізовщикам інструменти для створення будівельних конструкцій при мінімальних затратах, дозволяючи також створювати власні інструменти проектування;
- зв'язок з виробництвом сталевих конструкцій – взаємодією Revit і Advance Steel налагоджується і оптимізується робочий процес BIM починаючи від проектування до виготовлення сталевих конструкцій;
- ОВК – проектування і випуск документації – проектування і моделювання складних систем повітроводів і трубопроводів з різними компонентами в них;
- проектування і документування електричних систем – проектування, моделювання електричних систем і створення відповідної проектної документації, у тому числі забезпечення можливості відстеження електричних навантажень по всій мережі електроживлення;
- проектування сантехнічних систем і розробка документації – проектування водопровідно-каналізаційної системи з похилими трубами, компонування системи трубопроводів і розробка проектної документації;
- деталювання бази даних виробника MEP – дозволяє створити в Revit готові для виготовлення моделі деталей і компонентів, забезпечує можливість моделювати і координувати компоненти з УД 400;
- інтеграція Insight – дозволяє оптимізувати характеристики будівель за рахунок централізованого доступу до даних про продуктивність і поліпшеного механізму аналізу;

- перетворення служби бази даних виробника – інструмент «Проект для виробника» дозволяє перетворити елементи моделі рівня проекту в елементи з рівнем деталізації для будівництва;
- документація для виробника – дозволяє ефективніше описувати компонування моделі, документування, складання специфікації і відзначення елементів з бази даних виробника;
- будівельне моделювання – більш повне розкриття проектного задуму, закладеного в моделі, дозволяє здійснювати поділ шарів стін і заливок бетону та маніпулювання ними, а також забезпечує підготовку робочих креслень для виготовлення конструкцій;
- координація будівельних робіт – забезпечується вивантаження моделей Revit в BIM 360 Glue і їх синхронізація з BIM 360, а також перегляд та відправка наборів точок за допомогою веб-інтерфейсу BIM 360 Glue або застосування додатків BIM 360 Layout для iPad;
- взаємодія з Navisworks – дає можливість відкрити координаційну модель Navisworks безпосередньо в Revit для узгодження проектування з роботою проектних груп, якщо вони використовують різне ПО;
- імпорт і експорт 2D-листів – забезпечує перенесення файлів AutoCAD між Revit і іншими програмними середовищами і підготовка підшивки з характеристиками для експорту в системи виробників, субпідрядників та інших зацікавлених осіб у форматі, який підходить до їхніх систем;
- деталізація для забезпечення технологічності будівництва – додавання в проектні моделі необхідних деталей для забезпечення відповідності всім вимогам з боку підрядних організацій;
- виробництво будівельних конструкцій – забезпечується зв'язок з Advance Steel для деталювання і виготовлення сталевих

конструкцій, що прискорює процес від проектування до отримання сталевих конструкцій;

- виробництво будівельних конструкцій для інженерних систем – перетворення елементів моделі рівня деталізації проекту в елементи рівня бази даних виробника для деталізації координації, виготовлення і монтажу.

Можливості Revit найкраще підходять до реалізації задач, що пов'язані з будівництвом. Він також надає можливість інтеграції доповнень розробників через програмний інтерфейс .NET API, який у свою чергу дозволяє програмувати будь-якою мовою програмування, яка підтримується фреймворком .NET: VB.NET, C# та C++/CLI [9].

AutoCAD – система автоматизованого проектування, призначене для створення точних дво- та трьохвимірних моделей та креслень. Використовується в багатьох виробничих сферах та галузях діяльності людини: архітектура, машинобудівництво, побудова будівель, мостів, трубопроводів дизайн інтер'єру, проведення електродротів тощо. Основними можливостями даного редактора є наступні:

- доступність на всіх пристроях – перегляд, редагування, створення малюнків та моделей на мобільних пристроях, Інтернеті на комп'ютерах;
- підтримка хмарного сховища – отримання доступу до моделей з будь-якого місця за допомогою хмари Autodesk;
- палітра блоків – ефективне встановлення блоків зі списку, що недавно використовувався, із візуальними галереями;
- швидке вимірювання – відображення усіх вимірювань на кресленні простим наведенням курсором миші;
- розширене порівняння рисунків – порівняння двох версій малюнка, не залишаючи поточного вікна;
- налаштування текстів – створення одно- або багаторядкового тексту як єдиний текстовий об'єкт;

- таблиці – створення таблиць із даними та символами в рядках та стовпцях, застосування формул та зв'язок із електронними таблицями Microsoft Excel;
- хмара ревізії – швидке визначення оновлень у малюнку;
- зв'язок даних – одночасні оновлення у таблиці Microsoft Excel та таблицею в малюнку;
- вилучення даних – витягування інформації з об'єктів, блоків та атрибутів, включаючи інформацію про малюнок;
- експрес-інструменти – для підвищення продуктивності використовується колекція інструментів швидкого доступу;
- моделювання твердих тіл, поверхні та сіток – створення реалістичних 3D-моделей, використовуючи комбінацію інструментів для моделювання твердих тіл, поверхонь та сіток;
- тривимірна навігація – використовуються інструменти 3D-перегляду та навігації для обернення, повороту, прогулянки та обльоту навколо 3D-моделі;
- візуальні стилі – для керування відображенням країв, освітлення та затінення моделі;
- візуалізація – застосування освітлення та матеріалів, щоб надати моделям реалістичного вигляду;
- хмарна візуалізація – візуалізація моделей в Інтернеті для зменшення обчислювальних затрат;
- документація моделі – генерація двовимірних малюнків, які включають основу, проекції, перерізи та деталі з 3D-моделей;
- використання PDF та DGN файлів для обміну або повторного використання даних;
- посилання на інші малюнки;
- набір аркушів – перегляд, керування, малювання та доступ до декількох рисунків у вигляді набору аркушів;
- імпорт моделей з інших програм;

- географічне розташування та онлайн-карти;
- технологія TrustedDWG – технологія, що попереджає про можливу несумісність, коли файл останній раз не зберігався програмним забезпеченням Autodesk;
- налаштування графічного інтерфейсу для покращення доступності та зменшення кількості кроків для частих завдань;
- безпечне завантаження – обмеження безпеки для запуску виконувальних файлів для захисту від зловмисного виконаного коду;
- макроси – запис дій та вхідних значень, які потім можна відтворити;
- перевірка стандартів САПР – визначення та відстеження стандартів САПР для підтримки стилів для шарів, лінійних типів, тексту та розмірів.

Можливості та функції AutoCAD також добре підходять для вирішення задач, що пов'язані з будівництвом. Для розроблення доповнень для редактора AutoCAD надаються такі засоби: середа програмування ObjectARX з програмним інтерфейсом, написаним об'єктно-орієнтованою мовою C++, програмний інтерфейс AutoCAD .NET API, повна середа розроблення Visual LISP та інтерфейс ActiveX [10].

Програмне забезпечення для 3D-моделювання від компанії Autodesk дозволяє не стримувати уяву при реалізації такого роду завдань, але всі вони є достатньо дорогими. Ліцензійна підписка коштує від 1 545 доларів за рік [11], що дозволяє користування будь-яким з цих продуктів представниками корпоративного сектору та середнього бізнесу. Не зважаючи на те, що кожний з цих продуктів надає можливість розроблювати та інтегрувати власні доповнення, це можливо лише при придбанні ліцензії та завантаженні SDK.

Blender – програмне забезпечення з відкритим кодом для 3D-моделювання. Незважаючи на це, він підтримує весь 3D-конвеєр:

моделювання, такелаж, анімація, візуалізація, композиція, відстеження руху, редагування відео. Він містить у собі також 2D конвеєр анімації.

Це програмне забезпечення [12] дає можливість просунутим користувачам використовувати API Blender для сценаріїв Python з метою налаштування програми та написання спеціалізованих інструментів. Воно добре підходить для окремих користувачів та невеликих студій, які отримують прибуток від його єдиного конвеєра та чутливого процесу розробки.

Основними можливостями даного редактора є:

- візуалізація – дозволяє створювати вражаючі ефекти завдяки висококласному відстежувальному виробничому контуру Cycles;
- екран Blender для моделювання – сюди відносяться такі напрямки моделювання: скульптура, ретопологія, власне моделювання, криві тощо, які забезпечуються великим набором інструментів моделювання Blender;
- скульптура – цифрові інструменти, які забезпечують потужність та гнучкість, необхідну на декількох етапах цифрового виробничого конвеєра при моделюванні скульптури;
- анімація та такелаж – розроблений для анімації, Blender використовується для створення графіки короткометражних і художніх фільмів;
- жирний олівець – дозволяє проекти дизайну Story Art та 2D-дизайну передавати прямо у 3D-огляд Blender;
- VFX – забезпечує відстеження руху камери та об'єкта до маскування та композиції, їх можна «fix it in post»;
- симуляції – Blender має потужні інструменти симуляції різноманітних процесів завдяки наявності таких галузевих бібліотек, як Bullet і MantaFlow;
- конвеєр – інтеграція з декількома інструментами конвеєрів має вирішальне значення у багатьох процесах;

- редагування відео – інтегрований відеоредактор пропонує невелику кількість основних ефективних інструментів для редагування відео;
- сценарій – кожен інструмент доступний для сценаріїв та налаштування завдяки розширеному API Python;
- гнучкий інтерфейс – архітектура інтерфейсу Blender забезпечує повнофункціональне налаштування вікон та ярликів.

Blender, на відміну від редакторів Autodesk, є вільним та безкоштовним редактором з відкритою можливістю розроблювати та інтегрувати власні доповнення за допомогою мови програмування Python та програмного інтерфейсу Blender Python API. Також він крос-платформений, тобто працює під будь-якою операційною системою. В останні роки популярність даного редактора помітно зростає, він активно розвивається, постійно виходять нові оновлення з новою функціональністю та покращеним інтерфейсом користувача.

3D-редактор SketchUp – це одне з простих і безкоштовних рішень для 3D-моделювання в мережі. 3D-макети SketchUp є завжди доступними в мережі і завжди під рукою, за умови наявності доступу до Інтернет. Дуже простий в опануванні, не потребує залучення допомоги фахівців при проектуванні 3D-моделей.

Розглянемо основні функції та можливості цього редактора:

- взаємодія з іншими інструментами проектування;
- здатність до розширення з використанням Extension Warehouse;
- створення детальних звітів – генерування оцінок витрат, підсумування частин та інше з моделі;
- використання та проектування в віртуальній та доповненій реальності;
- імпорт файлів CAD – завантаження у редактор файлів з форматом CAD;

- експорт файлів CAD та PDF – вивантаження з редактору файлів у форматах CAD та PDF;
- створення багатосторінкових наборів презентацій;
- створення будівельних креслень;
- створення текстур;
- створення тривимірні та двовимірні моделі – побудова та відпрацювання дво- та трьохвимірних моделей;
- створення геолокаційних моделей;
- створення анімацій;
- експорт анімаційних відео будь-якого розміру;
- робота з модельованими плівковими камерами;
- імпорт, експорт та створення файлів у форматі IFC;
- створення світлових ефектів;
- керування шарами;
- використання хмарного сховища;
- використання оффлайн.

SketchUp розповсюджується безкоштовно, але має платну версію яка, у свою чергу, дешевше за редактори Autodesk – його ліцензія коштує 300 доларів за рік [13]. Він надає можливість розроблювати та інтегрувати власні доповнення за допомогою SketchUp SDK, який містить SketchUp C API, LayOut C API, інтерфейс для імпорту та експорту та повну документацію. Всі доповнення розроблюються за допомогою мови програмування C/C++ [14].

Окрім цих редакторів, є можливість створювати власні програмні забезпечення на основі рушіїв, таких як Unity3d та UnrealEngine.

Unity3d – це повноцінний ігровий рушій, призначений для створення ігор, дво- та трьохвимірних додатків під більшість популярних платформ, у тому числі для мобільних пристроїв, ігрових приставок та інших пристроїв, що призначені для комп’ютерних розваг.

Даний рушій інтегрований у редактор Unity, в якому одразу ж можна протестувати свою гру або додаток, не виходячи з нього.

Редактор Unity має наступні основні функції та можливості:

- анімація:
 - анімації, здатні на перенаправлення;
 - повний контроль над анімаційними вагами під час виконання;
 - події, які викликаються з відтворення анімації;
 - використання Blend Shapes для анімації обличчя;
- графіка:
 - глобальне освітлення у режимі реального часу на основі Enlighten;
 - фізично правильний шейдинг;
 - модульна система частинок, яка оснований на кривих та градієнтах;
 - інтуїтивні інструменти графічного інтерфейсу;
- оптимізація:
 - покращене профілювання пам'яті;
 - відключення оклюзії на основі Umbra;
 - групування активів;
 - підтримка рівнів деталізації;
 - зменшення розміру збірки;
 - багатопотокова система задач;
- дво- та трьовимірна фізика:
 - використання Box2D з багатим набором ефекторів, колайдерів та зчленувань;
 - підтримка NVIDIA PhysX 3.4;
- написання скриптів:
 - мови програмування – C# та JavaScript;
 - інтеграція з середовищем Visual Studio або MonoDevelop.

UnrealEngine також дозволяє розроблювати ігри та дво- та тривимірні додатки під будь-яку платформу, та він інтегрований у редактор Unreal Editor. Основними можливостями даного рушія є наступні:

- перетворення цілих сцен з інших 3D-редакторів та цілого ряду інших форматів за допомогою Datasmith;
- підтримка FBX, USD та Alembic;
- підтримка мови програмування Python;
- рендерінг та графіка – підсистема візуалізації, яка включає освітлення та затінення, матеріали та текстури, частинки та ефекти, та пост-обробку;
- візуалізація та моделювання волосся – рушій використовує робочий процес на основі пасм, щоб візуалізувати кожне окреме пасмо волосся фізично точним рухом, що дозволяє художникам імітувати та відтворювати сотні тисяч (а то й більше) моделей фотореалістичного волосся у режимі реального часу;
- UMG UI Designer – візуальний інструмент розробки графічного інтерфейсу;
- симуляція фізики – використовується для розрахунку колізій та симуляції фізичних акторів;
- анімація скелетної сітки на основі ключових кадрів анімації та морфних мішеней;
- використання візуальних ефектів Ніагара;
- ієрархічні рівні деталізації;
- використання активів та пакетної системи;
- використання різних систем координат;
- вбудовані доповнення.

Unity та UnrealEngine мають досить активну спільноту розробників, надають власні редактори для розроблення власних рішень та мають підтримку основних ОС – Windows, Linux та MacOS. Unity підтримує імпорт багатьох різних форматів, включаючи FBX, OBJ та відносно новий

формат glTF, а також формати деяких тривимірних редакторів, такі як Blender або продукти компанії Autodesk. UnrealEngine, у свою чергу, підтримує імпорт лише трьох форматів – FBX, OBJ та glTF. Редактори даних рушіїв надають власні засоби для редагування полігональних моделей оточення.

Однак детальне знайомство з функціями цих рушіїв та їх редакторів показало, що вони не підходять для вирішення задачі оптимізації розташування джерел освітлення приміщення.

Проведений аналіз можливостей програмного забезпечення для 3D-моделювання дозволив зробити висновки щодо доцільності створення доповнення до 3D-редактора.

На підставі аналізу можливостей проведено порівняння 3D-редакторів, що наведено в табл. 6.

Таблиця 6

Таблиця порівняння 3D-редакторів

		Autodesk (Maya, Revit, 3ds Max, AutoCAD)	SketchUp	Blender
Вартість ліцензії		1 545 доларів на рік і вище	300 доларів на рік	Безкоштовно
Засоби розроблення та інтеграції власних рішень		Autodesk SDK	SketchUp SDK	Blender Python API
Мова програмування		Python, MEL, C++/CLI, C#	C/C++	Python
Підтримка ОС	Windows	+	+	+
	Linux	–	–	+
	Mac OS X	+/-	+	+
Повнота документації для розробників		+/-	+/-	+/-
Можливість працювати з інструментами редактора		Невідомо	Можливо	Можливо

3.3. Структура програмного продукту

Розроблений програмний продукт є розширенням редактора тривимірних моделей Blender, що написаний мовою Python з використанням Blender Python API [15].

У архітектурі програми (рис. 6) можна виділити наступні модулі: модуль введення даних, модуль оптимізації, модуль перевірки рівня освітлення та модуль розміщення 3D-об'єктів.

Модуль введення даних дозволяє користувачеві вводити наступні дані:

- розмір приміщення (ширина, глибина та висота до світильника);
- розмір перегородки (довжина та ширина), кількість та її розміщення в кімнаті (початкова точка знаходиться на стіні, кінцева – на кінці довжини перегородки);
- тип робіт за точністю (висока, середня, низька, груба);

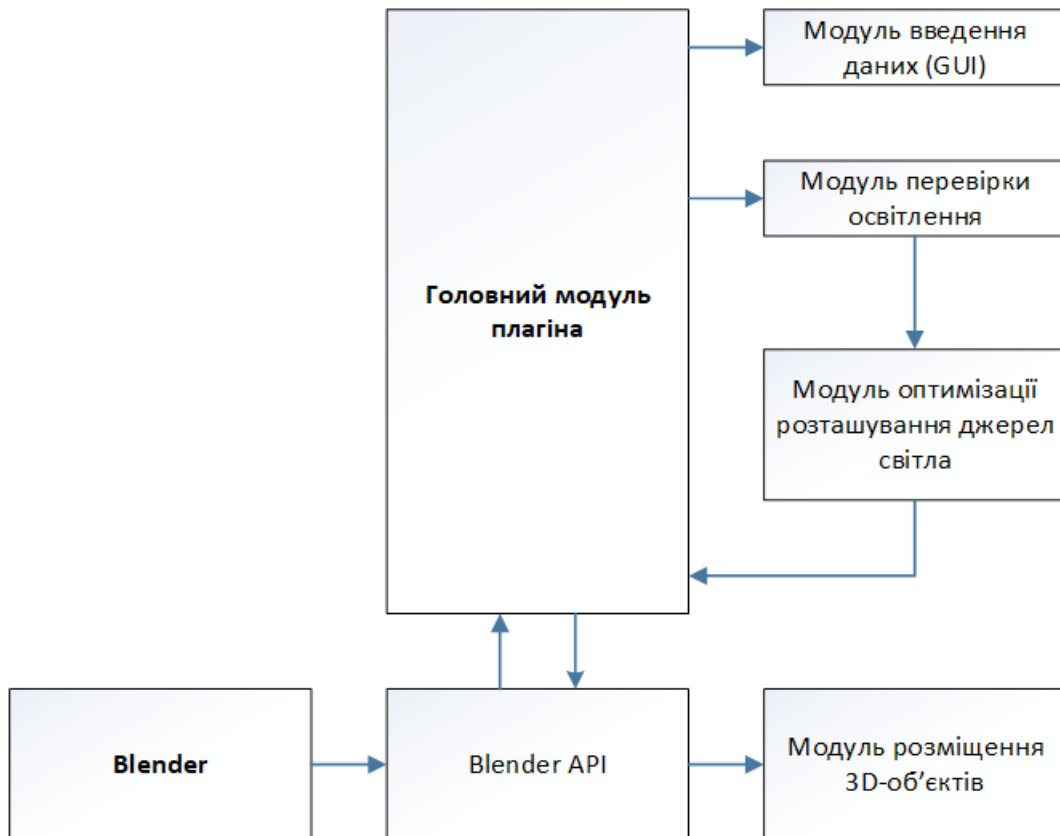


Рис. 6. Архітектура програмного продукту

- розмір вікон (ширина та висота) та їх кількість, причому допускається введення вікон різного розміру;
- висота до нижньої та до верхньої кромки вікна;
- тип світильників за потужністю (40 Вт, 60 Вт, 100 Вт, Т8: 36 Вт, Т5: 18 Вт) та їх кількість;
- розташування світильників (зліва, в глибину);
- розташування вікон (ширина вікон, відстань від початку стіни);
- на якій стіні розташовані вікна (фронтальна, ліва, права).

Також в цьому модулі створюються елементи графічного інтерфейсу, за допомогою яких є можливість реалізації зрозумілого процесу введення вхідних даних. Модуль запускається з головного модуля плагіна.

Результат роботи модуля – масив вхідних даних, які використовуються іншими модулями інноваційного програмного продукту. Діаграма класів модуля інтерфейсу (модуля введення даних) наведена на рис. 7.

В модулі перевірки рівня освітлення реалізовано алгоритм розрахунку рівня освітлення, що наведений у підрозділі 3.6. Для виконання своїх функцій даний модуль отримує дані з модуля введення даних. Перелік полів для зберігання та обробки вхідних даних наведено в табл. 7.

Таблиця 7

Перелік полів для зберігання та обробки вхідних даних

№ з/п	Назва поля та його тип	Коментар
1.	work: Work (from constants)	тип робіт
2.	width: float	ширина приміщення
3.	depth: float	глибина приміщення
4.	light: Light (from constants)	тип світильників
5.	light_height: float	висота від підлоги до світильника
6.	light_positions: list[tuple(float, float)]	розташування світильників (зліва, в глибину)
7.	w_top_height: float	висота до верхньої кромки вікна
8.	w_bottom_height: float	висота до нижньої кромки вікна

№ з/п	Назва поля та його тип	Коментар
9.	front_windows: list[tuple(float, float)]	розташування фронтальних вікон – відстань від початку стіни та ширина вікон
10.	left_windows: list[tuple(float, float)]	розташування лівих вікон (відстань від початку стіни, ширина вікон) або None, якщо стіна внутрішня
11.	right_windows: list[tuple(float, float)]	розташування правих вікон – None, якщо стіна внутрішня, або відстань від початку стіни та ширина вікон
12.	partitions: list[float]	перегородки, що ділять приміщення: - > 0 для перегородок, які спираються на фронтальну стіну; - < 0 для перегородок, які спираються на бокову стін

В модулі оптимізації реалізовано алгоритм оптимального розміщення джерел світла, що наведений у підрозділі 3.5. Для виконання своїх функцій даний модуль отримує з модуля перевірки рівня освітленості значення про необхідність оптимізації та дані, що були введені раніше.

Модуль розміщення 3D-об'єктів розташовує джерела освітлення після оптимізації автоматично, тобто користувачеві не має потреби розміщувати світильники та вікна самостійно. Для цього він з модуля оптимізації отримує нові дані про розташування світильників та вікон, їх кількість, а також рекомендований тип світильників та рекомендовані розміри вікон.

Blender Python API – бібліотека, або ж програмний інтерфейс додатку, для мови Python, що дозволяє створювати нові інструменти для Blender, нові елементи інтерфейсу користувача, а також доповнення до нього. Окрім цього, вона дозволяє змінювати дані 3D-об'єктів так само, як в інтерфейсі Blender.

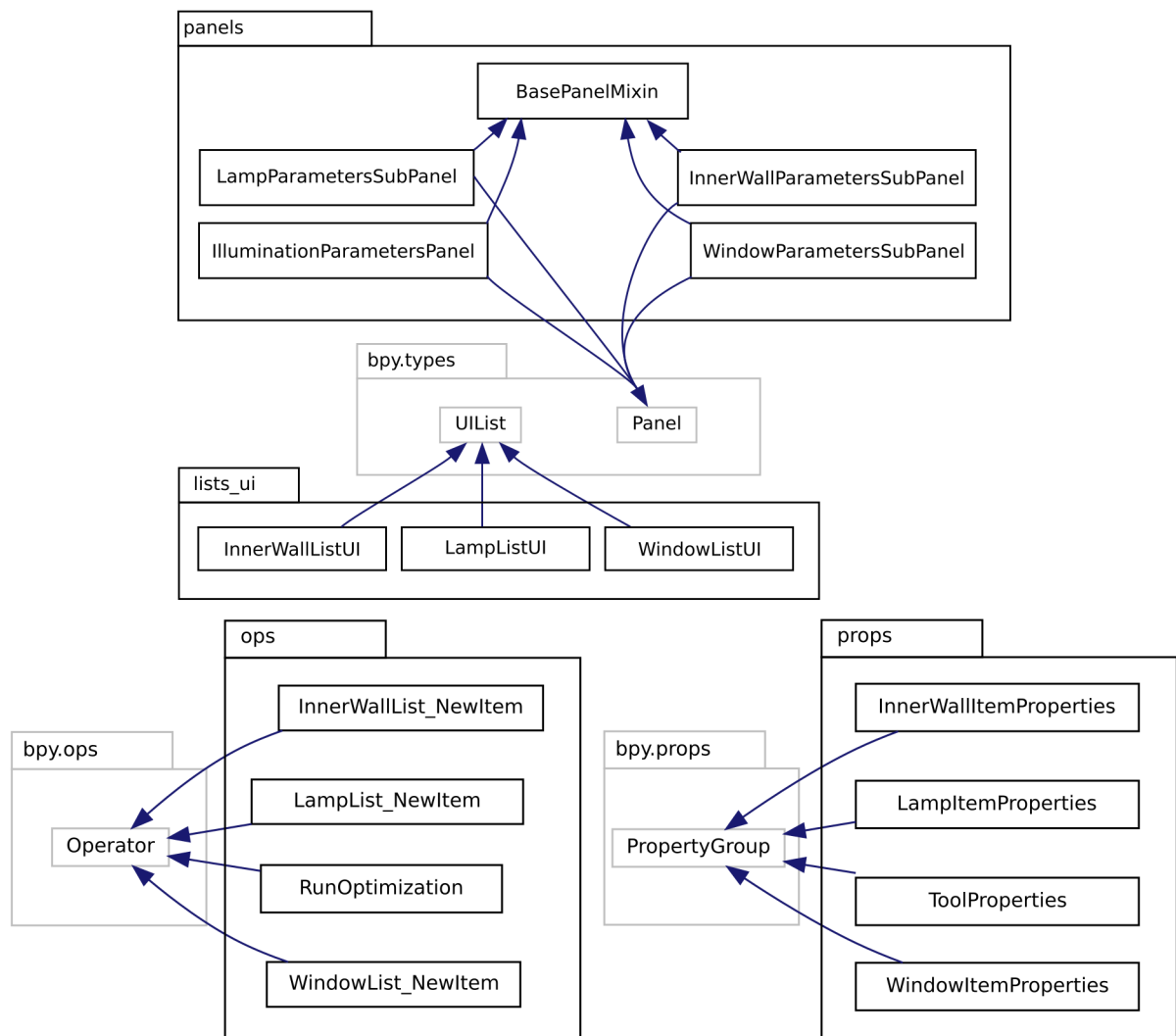


Рис. 7. Діаграма класів модуля інтерфейсу

3.4. Алгоритм введення даних

Алгоритм введення даних забезпечує можливість введення даних користувачем за допомогою графічного інтерфейсу (GUI). Зображення екранної форми для введення даних наведені на рис. 8 та рис. 9. Процес введення даних графічно розділено на декілька логічних блоків:

- основні параметри, де вводяться наступні дані:
 - тип робіт (за точністю);
 - розміри приміщення: ширина, глибина та висота від підлоги до світильника в метрах;
- внутрішні перегородки:
 - кількість та розмір перегородок: довжина та ширина в метрах;

- розташування (початкова точка на стіні, кінцева точка на кінці довжини перегородки);
- вікна, де вводяться:
 - кількість та розмір вікон: ширина, висота в метрах (допускається введення вікон різного розміру);
 - висота до нижньої кромки вікна в метрах;
 - висота до верхньої кромки вікна в метрах;
 - розташування (ширина вікна, відстань від початку стіни);
 - на якій стіні розташоване вікно (фронтальна, ліва, права);
- світильники, де вводяться:
 - тип світильників за потужністю (40 Вт, 60 Вт, 100 Вт, T8: 36 Вт, T5: 18 Вт), допускається вибір одного типу світильників;
 - початкова кількість світильників;
 - розташування світильників (зліва, в глибину в метрах).

▼ Lamps

Lights' elevation (in meters) 2.00

Lamp type: 60 W

💡	3.00	0.00
💡	3.00	2.00
💡	2.00	2.00

▶

Position: < 2.00 > 2.00

+ Add

- Delete

▼ Button

Do optimization

Рис. 8. Фрагмент екранної форми для введення даних

▼ General parameters

Work precision: Normal

Room size (in m, WxL): 3.00 4.00

▼ Inner Walls

Size (in m, WxL): 0.50 0.50

Position: 0.00 0.00

+ Add

- Delete

▼ Windows

Window elevations Bottom 0.50 Top 1.00

60 cm 60 cm Front → 2.00

Width: 60 cm Height: 60 cm

Wall offset 0.00 Wall: Front

+ Add

- Delete

▼ Lamps

Рис. 9. Екранна форма для введення даних

3.5. Алгоритм розрахунку рівня освітлення

В основі алгоритму розрахунку рівня освітлення використовується метод люмену та формула визначення рівня КПО, блок-схема алгоритму наведена на рис. 10.

Після початку роботи алгоритму, з програмного модуля введення даних отримуються вхідні дані, які необхідні для проведення перевірки рівня освітлення в приміщенні, який задано згідно з умовами задачі.

Перевірка рівня освітлення виконується за формулами (1.1) та (2.3) для штучного та природного відповідно.

За результатами перевірки виконується умова: якщо рівень освітлення відповідає нормам ДБН В.2.5-28:2018, то на екран про це виводиться повідомлення, якщо ні – виводиться повідомлення про низький рівень освітлення і запускається алгоритм проведення оптимізації розміщення джерел світла.

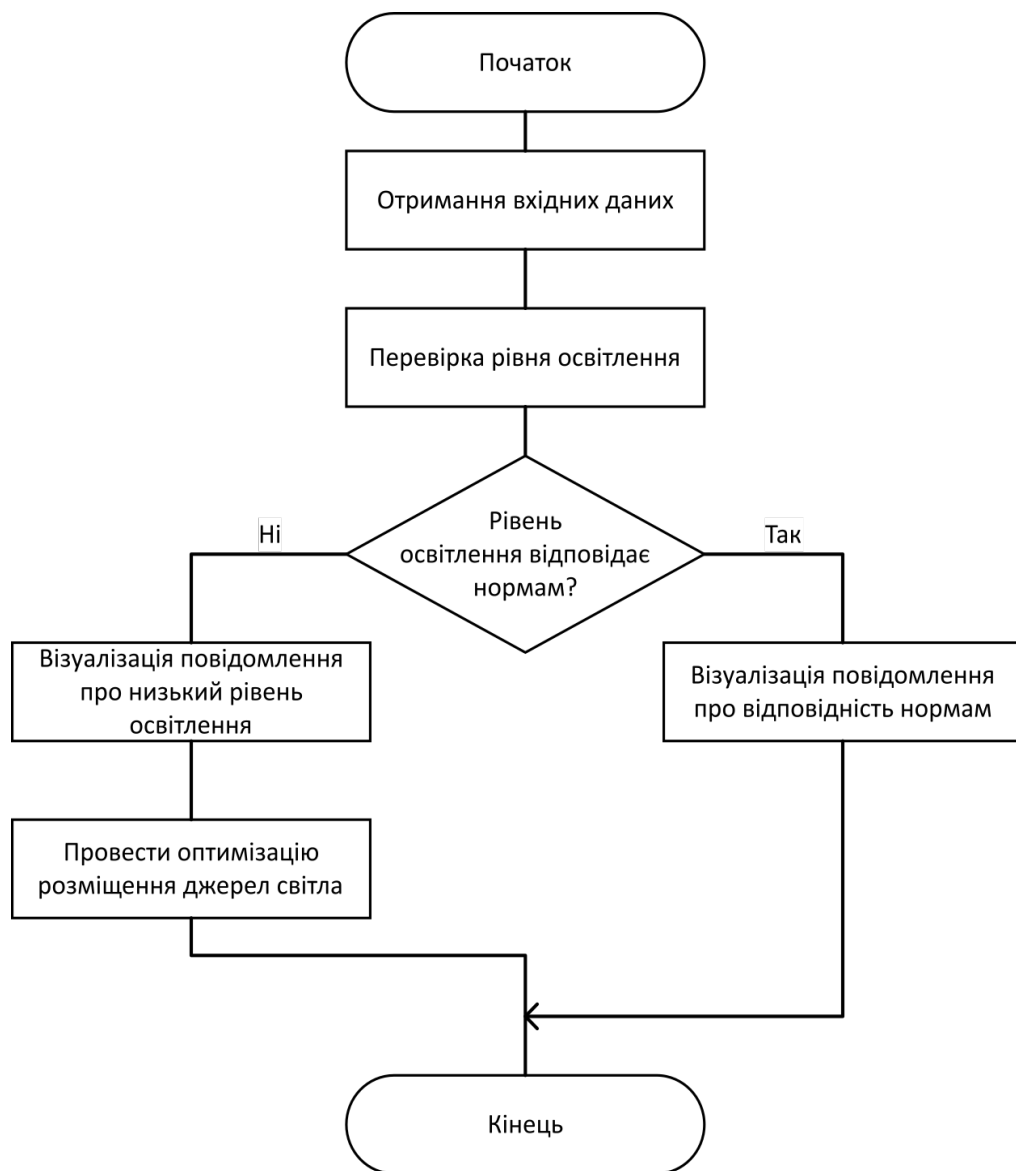


Рис. 10. Блок-схема алгоритму розрахунку рівня освітлення

3.6. Алгоритм оптимального розміщення джерел світла

Кроки даного алгоритму формалізовано подано на рис. 11.

Алгоритм оптимального розташування джерел світла запускається за умови, коли перевіркою встановлено, заданий рівень освітлення приміщення є низьким. Після старту роботи алгоритму виконується перевірка вхідних даних на наявність у приміщенні внутрішніх перегородок. Якщо в приміщенні перегородки відсутні, площа приміщення розбивається на квадрати. Кожний виділений квадрат має площу поверхні, яка може бути освітлена джерелом світла вибраної потужності (в даному

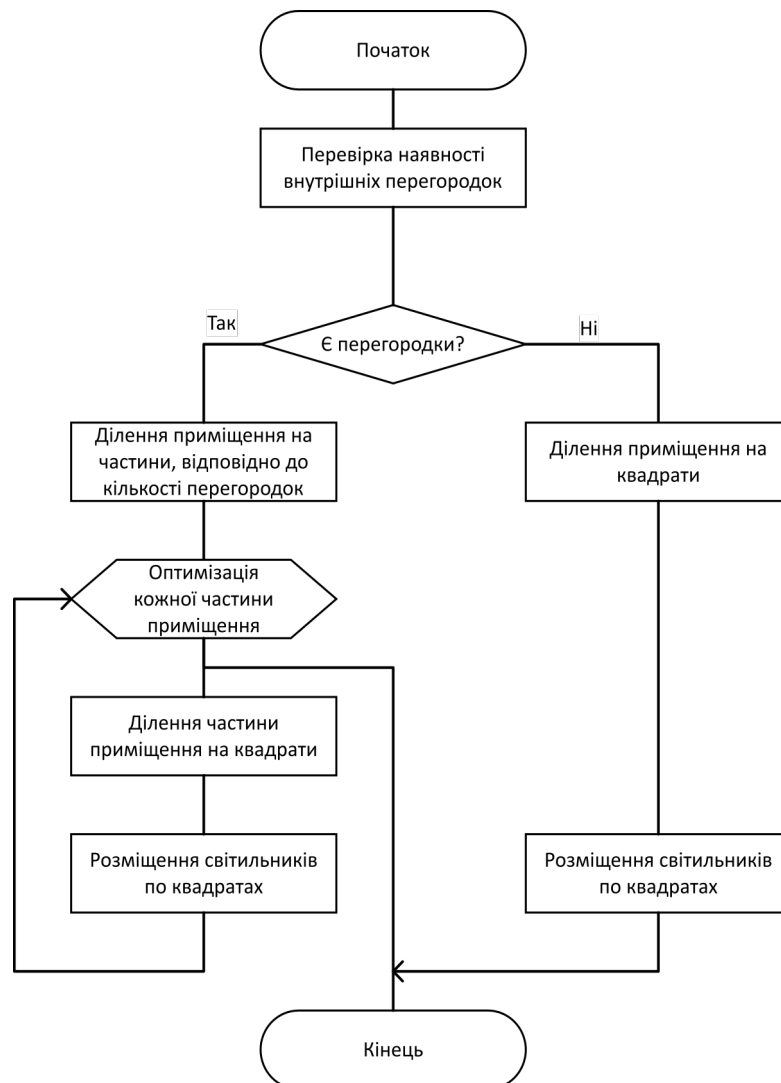


Рис. 11. Блок-схема алгоритму оптимального розміщення джерел світла

випадку маються на увазі світильники). Тобто кожне джерело світла у моделі приміщення має освітлювати квадратну ділянку. Таким чином, приміщення буде рівномірно освітлено, а рівень освітлення відповідатиме нормам стандарту. Далі програма розташовує джерела світла по таких квадратах. Вибір потужності джерела світла виконується автоматично.

Якщо в приміщенні присутні перегородки, то приміщення ділиться на частини та для кожної з цих частин виконуються операції з розбивки площі частини приміщення на квадрати з подальшим розташуванням джерел світла по квадратах.

3.7. Висновки до розділу 3

В даному розділі було виявлено основні вимоги до розроблюваного інноваційного програмного продукту, серед яких головними є:

- надавати можливість на підставі введених вхідних даних перевірити рівень освітлення в приміщенні на відповідність нормам стандарту ДБН В.2.5-28:2018;
- виконати процес оптимізації розташування джерел світла в приміщенні;
- відобразити в автоматичному режимі оптимальне розташування джерел освітлення.

Було виконано порівняння популярних редакторів тривимірних моделей стосовно засобів, які вони надають для створення розширень. За результатами порівняння обрано редактор Blender – буде створено розширення на основі його Python API.

На підставі основних вимог та вибраного 3D-редактора розроблено та наведено основні алгоритми та структуру інноваційного програмного продукту.

4. ТЕСТУВАННЯ ІННОВАЦІЙНОГО ПРОГРАМНОГО ПРОДУКТУ ТА АНАЛІЗ РОЗРОБЛЕНОГО СПОСОБУ

4.1. Методика тестування

Об'єктом тестування є розроблений програмний продукт та спосіб оптимізації розташування джерел освітлення в приміщенні.

Головною метою тестування є перевірка відповідності розробленого програмного продукту вимогам, які наведені у підрозділі 3.1.

Обсяг тестування містить:

- перевірку вимог до інноваційного програмного продукту;
- перевірку складу програмного продукту;
- перевірку процесу встановлення програмного забезпечення;
- перевірку роботи основних функцій програмного продукту.

На етапі тестування проводиться перевірка працездатності інноваційного програмного продукту відповідно до вимог, що наведені у підрозділі 3.1. Використані для тестування набори даних наведені у наступному пункті.

Обмеженнями в умовах проведення тестування є:

- відсутність вимог на заміну однієї програмної компоненти іншою, з аналогічними або розширеними функціями;
- відсутність вимог на заміну складу вхідних даних, зміну вхідних та вихідних документів;
- повторення окремих етапів випробувань не більше трьох разів.

Тестування інноваційного програмного продукту відбувається шляхом послідовного виконання дій, визначених даною методикою. Здійснюється тестування правильності реалізації алгоритму оптимального розміщення джерел світла.

Послідовність виконання тестування:

1. Запуск програмного модуля інноваційного програмного продукту.

На екрані з'явиться форма для введення вхідних даних та вікно зображення моделі приміщення.

2. Введення вхідних даних:

2.1. основні параметри, де вводяться наступні дані:

- тип робіт (за точністю);
- розміри приміщення: ширина, глибина та висота від підлоги до світильника;

2.2. внутрішні перегородки:

- кількість та розмір перегородок: довжина та ширина в метрах;
- розташування (початкова точка на стіні, кінцева точка на кінці довжини перегородки);

2.3. вікна, де вводяться:

- кількість та розмір вікон: ширина, висота в метрах (допускається введення вікон різного розміру);
- висота до нижньої кромки вікна в метрах;
- висота до верхньої кромки вікна в метрах;
- розташування (ширина вікна, відстань від початку стіни);
- на якій стіні розташоване вікно (фронтальна, ліва, права);

2.4. світильники, де вводяться:

- тип світильників за потужністю (40 Вт, 60 Вт, 100 Вт, T8: 36 Вт, T5: 18 Вт), допускається вибір одного типу світильників;
- початкова кількість світильників;
- розташування світильників (зліва, в глибину в метрах).

3. Запуск модуля перевірки рівня освітлення, в якому виконуються:

- 3.1. перевірка на відповідність заданого освітлення будівельним нормам (доцільність оптимізації);
- 3.2. алгоритм оптимального розміщення джерел світла.
4. Виведення на екран сервісних повідомлень або результатів виконання алгоритму оптимального розміщення джерел світла.
5. Виведення на екран графічного зображення з оптимальним розміщенням джерел світла.

Окрім цього, для перевірки роботи основних компонент програмного забезпечення проводиться покриття юніт-тестами цих самих модулів за допомогою вбудованої в Python бібліотеки unittest, а також за допомогою тестових сценаріїв.

4.2. Створення тестових моделей приміщення

Для тестування використовуються наступні тестові набори даних:

- Набір 1: прямокутне житлове приміщення розміром (Ш x Г x В) 3,5 x 5,5 x 2,5 м з одним вікном розміром (Ш x В) 1,2 x 1,5 м на зовнішній стіні (ширина). Посередині кімнати розташований один світильник потужністю в 100 Вт. Задача перевірити відповідність природного і штучного освітлення відповідно до ДБН В.2.5-28:2018 “Природне та штучне освітлення”, затверджені наказом Міністерства регіонального розвитку, будівництва та житлово-комунального господарства України від 03.10.2018 та чинні від 01.03.2019.
- Набір 2: квадратна офісна кімната розміром (Ш x Г x В) 5,5 x 5,5 x 2,5 м з двома вікнами розміром (Ш x В) 1,2 x 1,5 м на протилежних зовнішніх стінах. Посередині кімнати розташований два світильники потужністю в 60 Вт. Задача перевірити відповідність природного і штучного освітлення відповідно до ДБН В.2.5-28:2018 “Природне та штучне освітлення”, затверджені наказом Міністерства регіонального розвитку, будівництва та

житлово-комунального господарства України від 03.10.2018 та чинні від 01.03.2019.

4.3. Створення тестових сценаріїв

Для перевірки роботи окремих компонентів програмного забезпечення, а також працездатності елементів графічного інтерфейсу здійснюються послідовні кроки тестового сценарію. Приклад тестового сценарію перевірки графічного інтерфейсу наведений у табл. 8.

Таблиця 8

Тестовий сценарій перевірки графічного інтерфейсу

№ з/п	Кроки тестового сценарію	Результат перевірки
1.	Встановлення плагіну не викликає помилку	пройдено
2.	Запуск плагіну	пройдено
3.	Введення вхідних даних	
3.1.	Розкриття панелей надає доступ до інших елементів інтерфейсу	пройдено
3.2.	Введення основних даних	
3.2.1.	Тип робіт (за точністю)	
3.2.1.1.	Висока	пройдено
3.2.1.2.	Середня	пройдено
3.2.1.3.	Низька	пройдено
3.2.1.4.	Груба	пройдено
3.2.2.	Розміри приміщення	
3.2.2.1.	Ширина	пройдено
3.2.2.2.	Глибина	пройдено
3.2.2.3.	Висота від підлоги до світильника	пройдено
3.3.	Введення відомостей про перегородки	
3.3.1.	Розмір	
3.3.1.1.	Ширина	пройдено
3.3.1.2.	Довжина	пройдено
3.3.1.3.	Кількість	пройдено
3.3.2.	Розташування	
3.3.2.1.	Початкова точка на стіні	пройдено

№ з/п	Кроки тестового сценарію	Результат перевірки
3.3.2.2.	Кінцева точка на кінці довжини перегородки	пройдено
3.4.	Введення відомостей про вікна	
3.4.1.	Кількість	пройдено
3.4.2.	Розмір	
3.4.2.1.	Ширина	пройдено
3.4.2.2.	Висота	пройдено
3.4.3.	Висота до нижньої кромки вікна, метрів	пройдено
3.4.4.	Висота до верхньої кромки вікна, метрів	пройдено
3.4.5.	Розташування	
3.4.5.1.	Ширина вікна	пройдено
3.4.5.2.	Відстань від початку стіни	пройдено
3.4.6.	На якій стіні розташоване вікно	
3.4.6.1.	Фронтальна	пройдено
3.4.6.2.	Ліва	пройдено
3.4.6.3.	Права	пройдено
3.5.	Введення відомостей про світильники	
3.5.1.	Тип світильників (за потужністю)	
3.5.1.1.	40 Вт	пройдено
3.5.1.2.	60 Вт	пройдено
3.5.1.3.	100 Вт	пройдено
3.5.1.4.	T8: 36 Вт	пройдено
3.5.1.5.	T5: 18 Вт	пройдено
3.5.2.	Кількість	пройдено
3.5.3.	Розташування	
3.5.3.1.	Зліва приміщення	пройдено
3.5.3.2.	В глибину приміщення	пройдено
3.6.	Додавання у список з'являється новостворений елемент списку для	
3.6.1.	Вікон	пройдено
3.6.2.	Ламп	пройдено
3.6.3.	Внутрішніх перегородок	пройдено
3.7.	Зміна даних в елементі списку призводить до оновлення елемента новими даними для	
3.7.1.	Вікон	пройдено

№ з/п	Кроки тестового сценарію	Результат перевірки
3.7.2.	Ламп	пройдено
3.7.3.	Внутрішніх перегородок	пройдено
3.8.	Натискання кнопки видалення призводить до видалення вибраного елемента списку для	
3.8.1.	Вікон	пройдено
3.8.2.	Ламп	пройдено
3.8.3.	Внутрішніх перегородок	пройдено
3.9.	Натискання кнопки додавання призводить до додавання нових даних до списку для	
3.9.1.	Вікон	пройдено
3.9.2.	Ламп	пройдено
3.9.3.	Внутрішніх перегородок	пройдено
4.	Натискання кнопки “Почати оптимізацію” призводить до виконання програмного модуля перевірки рівня освітлення в приміщенні	пройдено
5.	Обробка результатів роботи програмного модуля перевірки освітлення в приміщенні	
5.1.	Виведення повідомлення про відповідність освітлення приміщення нормам стандарту ДБН В.2.5-28:2018	пройдено
5.2.	Виведення повідомлення про низький рівень освітлення приміщення	пройдено
5.3.	Запуск програмного модуля оптимізації розташування джерел світла	пройдено
6.	Візуалізація результатів оптимізації – розташування джерел світла в приміщенні в формі 3D зображення	пройдено

4.4. Аналіз отриманих результатів

На етапі тестування було проведено послідовне виконання дій, визначених методикою тестування, за допомогою яких було перевірено працездатність програмного продукту та його основних функцій відповідно до вимог. Для тестування були зазначені два набори тестових

даних, в яких вказано тип світильників, їх кількість, розміри, тип та форму кімнати, а також кількість та розміри вікон.

На основі отриманих результатів можна зробити висновок, що метод оптимального розміщення джерел світла є ефективним для проектування ергономічних приміщень та можна надати рекомендації щодо приведення освітлення приміщень до вимог будівельних норм.

Також у ході проведення робіт виявлено, що спосіб оптимального розміщення джерел світла реалізований у програмному забезпеченні у повному обсязі, програмний продукт працював у штатному режимі, збої та відмови в роботі ПЗ не виявлено.

4.5. Висновки до розділу 4

У даному розділі була розроблена методика тестування інноваційного програмного продукту, метою якою є перевірка працездатності програми та відповідність до вимог, що були зазначені раніше. Для тестування було розроблена послідовність дій, які необхідно виконати, та створено набори тестових даних.

Аналіз отриманих результатів показує, що розроблений спосіб оптимального розміщення джерел світла є ефективним та реалізований у програмному продукті у повному обсязі, само програмне забезпечення працювало штатно без збоїв та відмов у роботі.

5. ІННОВАЦІЙНІСТЬ ПРОГРАМНОГО ПРОДУКТУ

5.1. Актуальність проблеми, що вирішує програмний продукт

Неефективне використання джерел світла (як природного, так й штучного) є актуальною проблемою, що стосується не тільки нових будинків, а й вже збудованих. На сьогодні не існує такого програмного забезпечення, який дозволив би користувачеві, який захоче купити квартиру, спочатку оцінити, наскільки буде комфортно в ній знаходитись протягом року. Також немає такого програмного продукту, який дав би можливість проєктувальникам та архітекторам подивитись прототип приміщення, оцінити його ергономічність в сенсі освітлення та у разі необхідності зробити розміщення джерел світла найбільш оптимально.

Тому виникають наступні проблеми через відсутність програмного продукту:

- відсутність оцінки ергономічності приміщення на етапі проєктування;
- відсутність оцінки “комфортабельності” приміщення;
- високі витрати на реалізацію проєкту;
- мало інформації про квартиру;
- дискомфорт при знаходженні в квартирі;
- високі витрати на зміну обстановки на більш комфортну.

Окрім цього, через неефективне використання джерел світла виникають, особливо штучного, виникають ряд проблем зі здоров'ям та порушується сон у людини. А це призводить до нервових розладів, дратівливості та недосипання, що знижують продуктивність людини. Також виростають витрати на спожиту електроенергію та скорочується строк служби ламп.

При надмірній кількості природного світла в квартирі, в залежності від пори року, стає дуже світло та дуже жарко. Також виникає велика кількість відблисків від поверхонь, наприклад від паркету чи ламінату.

Ієрархія проблем наведена у вигляді дерева на рис. 12.

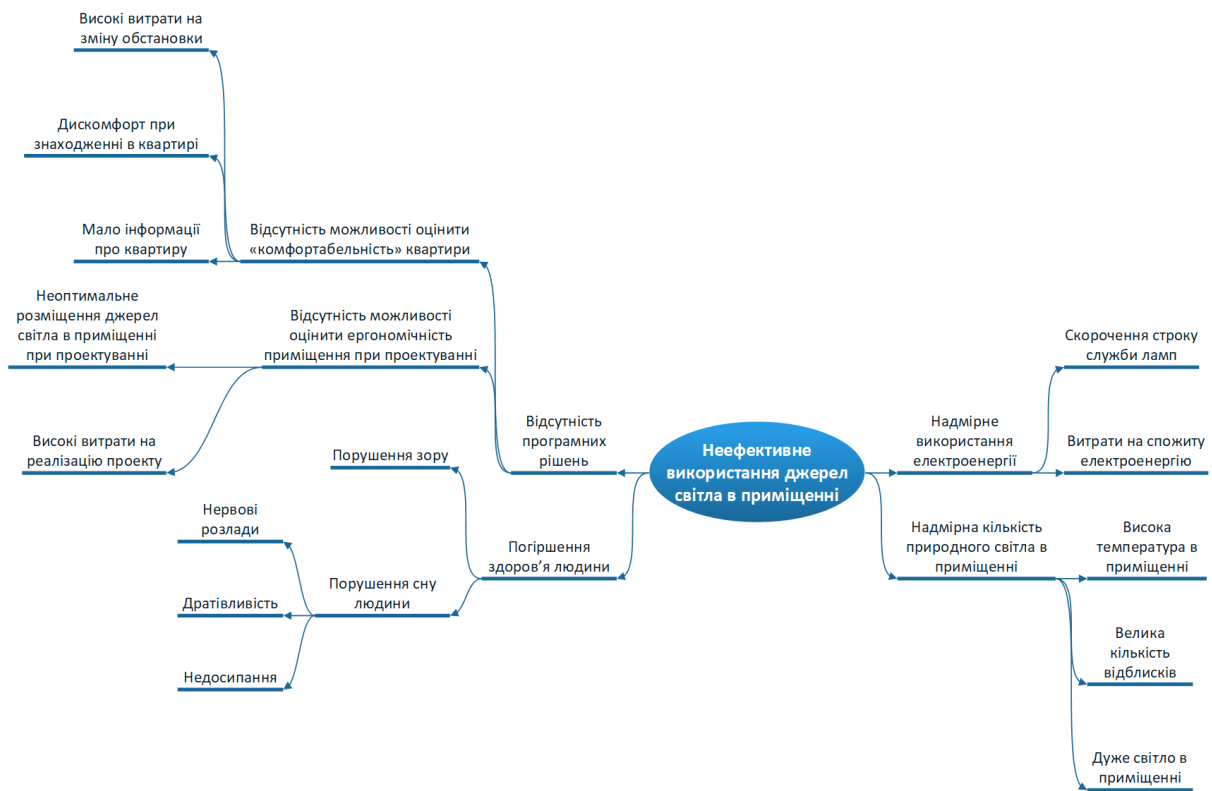


Рис. 12. Дерево проблем

5.2. Зацікавлені сторони

Даний програмний продукт, що розв'язує проблему, має декілька зацікавлених сторін.

Найбільш впливовими сторонами є забудовник та проектувальник. Вони витрачають великі кошти та час на те, щоб реалізувати проект, тобто побудувати будинок та приміщення.

Для забудовника важливий зручний спосіб відображення проекту, який необхідно реалізувати, адже рівень освітленості приміщення залежить не тільки від матеріалу стін, полу та стелі, а й від оптимального розміщення джерел світла, як звичайних ламп, так й вікон.

Для проектувальника важлива наявність можливості оцінити ергономічність в сенсі освітлення, щоб найбільш оптимально розмістити всі джерела світла. Адже від нього залежить якість побудованого приміщення та його конкурентоспроможність на ринку. Приміщення, в

якого рівень освітлення не відповідає нормам, матиме низьку конкуренцію, не зацікавить покупців та внаслідок компанія-забудовник матиме лише збитки.

Наступними зацікавленими сторонами із середнім впливом є маркетологи та покупці квартир. Маркетологам необхідна можливість показати покупцям вже готову квартиру з різними варіантами обстановки, не виходячи з відділу продажу, а також розрекламувати її, надаючи конкурентних переваг. Адже від них залежить, наскільки швидко продадуться квартири в новому житловому комплексі.

Покупцям квартир важливо оцінити “комфортабельність” квартири, наскільки комфортно та зручно їм буде в ній проживати на довгий період часу. Вони вкладають багато коштів не тільки на її покупку, а й на ремонт, щоб зробити обстановку більш комфортною. Дана сторона є важливою, так як від неї залежить майбутнє компанії-забудовника та його репутації на ринку.

Щоб привабити ці сторони необхідно проводити презентації програмного продукту для представників зацікавлених осіб, для покупців можна розмістити рекламу в магазинах, що містять такі ж додатки. Окрім цього, необхідно брати участь в спеціалізованих виставках та форумах.

В табл. 9 зведено усі групи зацікавлених осіб, їх інтерес та вплив, а також стратегії їх приваблення.

Таблиця 9

Зацікавлені сторони

Зацікавлена сторона	Інтерес	Вплив	Стратегії приваблення
Забудовник	Зручний спосіб відображення проєкту, який необхідно реалізувати	Високий	Проведення презентацій продукту для представників зацікавлених осіб. Участь в спеціалізованих виставках та форумах.

Зацікавлена сторона	Інтерес	Вплив	Стратегії приваблення
Проектувальник/архітектор	Спосіб побудови прототипу для оцінки ергономічності приміщення в сенсі освітлення	Високий	Проведення презентацій продукту для представників зацікавлених осіб. Участь в спеціалізованих виставках та форумах.
Маркетологи	Можливість показати готову квартиру та надати їй конкурентності	Середній	
Покупці квартир	Можливість оцінити "комфортабельність" квартири	Середній	Проведення презентацій продукту. Участь в спеціалізованих виставках та форумах. Розміщення реклами в онлайн-магазинах.

5.3. Основні характеристики програмного рішення

Визначивши основні проблеми, що існують на даний момент, можна описати кінцевий програмний продукт, який має їх вирішувати. Дане програмне забезпечення буде реалізовувати метод оптимального розміщення джерел світла, зазначений у попередніх розділах. Цей метод дозволить оптимально розмістити джерела світла так, щоб була висока ефективність використання як штучного, так й природного освітлення. Впровадження такого програмного продукту буде надзвичайно корисним для забудовників та проектувальників, оскільки він дозволить спроектувати приміщення з відповідними нормами щодо освітлення. І таким чином, у людей, які будуть знаходитись в таких приміщеннях у короткий або довгий період часу, зменшаться або зникнуть проблеми зі здоров'ям та нормалізується сон. Також зменшаться витрати на споживання електроенергії.

Очевидно, що клієнтами даного програмного забезпечення є забудовники та проектувальники. Співпраця буде побудована на моделі співробітництва бізнесу для бізнесу, тобто B2B.

Даний програмний продукт є по суті надбудовою над 3D-редактором, тому він повинен легко вбудовуватись з існуючими рішеннями 3D-моделювання.

5.4. Конкурентні переваги рішення

Оскільки сфера побудови приміщень різного типу є не новою, реалізації проектування таких кімнат, у тому числі й розміщення джерел світла, існує немало. Наразі існує декілька конкурентів, які розраховують рівень освітлення в приміщенні, і навіть моделюють його, однак вони не займаються оптимізацією розміщення джерел світла. І це являється основною конкурентною перевагою програмного продукту, що пропонується.

Також конкурентними перевагами даного програмного продукту є:

- можливість оцінити ергономічність прототипу в сенсі освітлення;
- можливість оцінити “комфортабельність” приміщення;
- зменшення споживання електроенергії;
- зменшення кількості ламп для штучного освітлення завдяки їх оптимальному розміщенню.

5.5. Клієнти. Сегменти ринку споживання

Як зазначалося вище, основними клієнтами даного програмного продукту є забудовники та проектувальники. Основна задача забудовників – збудувати проект відповідно до стандартів та основних вимог безпек та надійності, спираючись на розроблений прототип від проектувальника. Задача проектувальника – спроектувати приміщення відповідно до стандартів, в тому числі до норм освітлення для конкретного типу приміщення. Окрім цього, цим двом клієнтам необхідно забезпечити

економію електроенергії та безпеку життя та здоров'я людей, що будуть проживати або працювати в приміщеннях.

На даний момент проектуються і будуються багато різних приміщень, які можна сегментувати за типом. Наприклад, можна поділити приміщення за такими типами: житлові, офісні, торгові, медичні, виробничі тощо. В Україні наразі активно проектуються та будуються житлові комплекси, в яких також можна розмістити офіси. Однак найбільш важливими приміщеннями для проектувальників є лише офіси в таких будинках, а також виробничі, медичні, торгові – ті, в яких потім майже неможливо змінити освітлення (тобто, змінити розміщення світильників або розмістити по-іншому вікна). Тому, найбільш перспективними компаніями, що займаються проектуванням та забудовою, є ті, що спеціалізовані на виробничих, медичних, торгових та офісних приміщеннях.

Крім цього, дані компанії можна сегментувати за їх розміром: малі, середні та великі. Найбільш потенційними клієнтами є середні та великі компанії, оскільки для малих даних програмний продукт буде лише підвищувати складність та вартість проекту, і він сам не буде рентабельним.

5.6. Унікальна ціннісна пропозиція

Ціннісна пропозиція – це пояснення того, як продукт розв'язує проблему. Її можна скласти за наступною формулою:

Ціннісна пропозиція = (Проблема + Рішення) / Продукт.

У дереві проблем було визначено цілу низку проблем, пов'язані із неефективним використанням світла, а у зацікавлених сторонах – очікування відповідних сторін від програмного продукту та їх вплив.

Забудовнику важливо мати зручний спосіб відображення проекту, який необхідно реалізувати, в той час як проєктувальникам та архітекторам

необхідний спосіб побудови прототипу для оцінки ергономічності приміщення в сенсі освітлення, а покупцям квартир важливо мати можливість оцінити “комфортабельність” квартири. Запропоноване рішення дозволяє задовольнити наведені вище вимоги перших двох зацікавлених сторін та розв’язати проблеми, що наведені в дереві проблем.

Отже, основною унікальною ціннісною пропозицією є розроблений метод оптимального розміщення джерел світла (в основному, штучного), перевагою якого є оптимізація розміщення джерел штучного світла, яке не проводиться ні в одного з конкурентів.

5.7. Доходи та витрати

Основним джерелом доходу є продаж повноцінних ліцензій на використання програмного продукту, а також надання розширеної технічної підтримки. Під повноцінною ліцензією мається на увазі надання ліцензії на постійне користування послугою або програмним продуктом. Платна розширена технічна підтримка, у порівнянні зі стандартною, скорочує термін надання відповіді на запити користувача та усунення дефектів.

Основні статті витрат складатимуть:

- проведення науково-дослідних робіт;
- оплата праці персоналу;
- утримання робочих місць персоналу та виробничої інфраструктури;
- придбання обладнання, ліцензій програмного забезпечення для розробки;
- податкові витрати.

Витрати на реалізацію проекту та прогнозовані прибутки за перший рік діяльності стартап-проекту зведені у таб. 10.

Витрати на реалізацію проекту, тис. доларів США

Найменування витрат	Місяці						
	1	2	3	4	5	6	7
Загальні витрати	5	5	4,5	4,5	4,5	4,5	4,5
ЗП	4	12	12	12	12	12	12
Підсумок витрат	9	17	16,5	16,5	16,5	16,5	16,5
Заплановані прибутки	—	—	—	—	—	—	—
Результат (без оподаткування)	–9	–17	–16,5	–16,5	–16,5	–16,5	–16,5
Найменування витрат	Місяці					Загальні результати	
	8	9	10	11	12		
Загальні витрати	4,5	4,5	4,5	4,5	4,5	55	
ЗП	13	12	12	12	12	137	
Підсумок витрат	17,5	16,5	16,5	16,5	16,5	192	
Заплановані прибутки	45	50	55	55	60	265	
Результат (без оподаткування)	27,5	33,5	38,5	38,5	43,5	89,5	

5.8. Бізнес-модель

Узагальнимо все написане вище та складемо бізнес-модель у вигляді канви.

Споживачі: компанії, що займаються проектуванням та забудовою нових будівель.

Проблема: відсутність програмних рішень; відсутність оцінки ергономічності приміщення на етапі проектування; відсутність оцінки “комфортабельності” приміщення; високі витрати на реалізацію проекту; погіршення здоров’я людини; надмірне споживання електроенергії; надмірна кількість природного світла в приміщенні.

Рішення: програмне забезпечення, що дозволяє оптимально розмістити джерела світла на етапі проектування шляхом розрахунку рівня

освітлення для відповідності нормам освітлення та безпосередньо оптимізації розміщення даних джерел світла.

Унікальна ціннісна пропозиція: розроблений метод оптимального розміщення джерел світла з метою підвищення ергономічності приміщення в сенсі світла та зменшення споживання електроенергії.

Потоки доходів: доходи від продажу ліцензій програмного продукту; доходи від підтримки програмного забезпечення.

Структура витрат: проведення науково-дослідних робіт; оплата праці персоналу; утримання робочих місць персоналу та виробничої інфраструктури; придбання обладнання, ліцензій програмного забезпечення для розробки; податкові витрати.

Також в канву бізнес-моделі включаються структурні блоки: прихована перевага (перевага, яку неможливо скопіювати або купити), ключові метрики (основні показники, що вимірюються), та канали (шляхи до користувачів).

Канали: відділи інтеграції компаній, що займаються проектуванням та забудовою нових будівель.

Ключові метрики: кількість проданих ліцензій.

Прихована перевага: відповідність до норм освітлення, зазначених в ДБН України.

Розроблену бізнес-модель наведено у зведеному вигляді в табл. 11. Її аналіз дозволяє зробити висновок, що запропонований проект, який реалізує описаний у дисертації метод оптимального розміщення джерел світла, має перспективи у своїй подальшій реалізації.

Канва бізнес-моделі

<p>Проблема</p> <p>відсутність програмних рішень</p> <p>відсутність оцінки ергономічності приміщення на етапі проектування</p> <p>погіршення здоров'я людини</p> <p>надмірне споживання електроенергії</p>	<p>Рішення</p> <p>програмне забезпечення, що дозволяє оптимально розмістити джерела світла (в основному штучні) на етапі проектування</p> <p>Ключові показники</p> <p>кількість проданих ліцензій</p>	<p>Унікальна ціннісна пропозиція</p> <p>розроблений метод оптимального розміщення джерел світла з метою підвищення ергономічності приміщення в сенсі світла та зменшення споживання електроенергії</p>	<p>Прихована перевага</p> <p>відповідність до норм освітлення, зазначених в ДБН України</p> <p>Канали</p> <p>відділи інтеграції компаній, що займаються проектуванням та забудовою нових будівель</p>	<p>Споживачі</p> <p>компанії, що займаються проектуванням та забудовою нових будівель</p>
<p>Структура витрат</p> <p>проведення науково-дослідницьких робіт</p> <p>оплата праці персоналу</p> <p>утримання робочих місць персоналу та виробничої інфраструктури</p> <p>придбання обладнання, ліцензій програмного забезпечення для розробки</p> <p>податкові витрати</p>		<p>Потоки доходів</p> <p>доходи від продажу ліцензій програмного продукту</p> <p>доходи від підтримки програмного забезпечення</p>		

5.9. Висновки до розділу 5

У даному розділі було проведено аналіз поточної ситуації у сфері будівництва приміщень, виявлено основні проблеми, пов'язані із світлом, та зведено їх у відповідне дерево проблем. Наряду з проблемами було виділено зацікавлені сторони, їх інтерес до продукту та ступінь впливу даних сторін на розв'язання проблем.

Як наслідок було запропоноване комерційне рішення з конкурентними перевагами, в основу якого лежить розроблений метод оптимального розміщення джерел світла з метою підвищення ергономічності приміщення в сенсі світла та зменшення споживання електроенергії. Дане рішення задовольняє інтереси зацікавлених сторін, а саме забудовників та проектувальників.

Було проведено аналіз майбутніх клієнтів та досліджено сегменти ринку споживання. На основі даного аналізу було визначено унікальну ціннісну пропозицію та зроблено прогнозування потенційних доходів та витрат на реалізацію стартап-проекту.

Також наведена бізнес-модель, що узагальнює всі проведені дослідження та аналіз доцільності реалізації даного програмного продукту.

ВИСНОВКИ

Метою даної магістерської дисертації була розроблення способу оптимального розміщення джерел світла в приміщенні, що дозволить підвищити ергономічність даного приміщення, а також зменшити споживання електроенергії.

Для досягнення даної мети було здійснено такі роботи:

- надано визначення видів освітлення (природне: бокове, верхнє, комбіноване; штучне: робоче, комбіноване, локалізоване, місцеве);
- визначено типи світла (пряме, непряме, віддзеркалене, дифузне);
- проведено аналіз методів розрахунку рівня освітлення в приміщенні. Це метод люмену, який крім всього іншого також дозволяє знайти необхідну кількість світильників, знаючи рівень освітлення приміщення, метод Сампнера, яким можна розраховувати рівень освітленості можливо, коли відомі відбивні властивості поверхні та кількість світлового потоку, що входить в простір, та поточковий метод, який може бути використаний тільки на точкових джерелах світла. У ході аналізу проведено узагальнення та порівняльний аналіз існуючих методів розрахунку рівня освітлення в приміщенні, яке дозволило обрати найбільш придатний до обчислень метод, а саме метод люмену;
- виконано розроблення способу розміщення джерел освітлення та поставлено задачу розроблення інноваційного програмного продукту, що реалізуватиме спосіб оптимального розміщення джерел освітлення та забезпечить проведення перевірки рівня освітлення в приміщенні, оптимізацію розміщення джерел освітлення, візуалізацію повідомлення про порушення вимог стандарту тощо;
- сформовані наступні гіпотези:
 - рівень освітлення можна обчислити, використовуючи метод люмену;

- обчислення можна проводити в реальному часі;
- за допомогою розрахунку рівня освітлення в приміщенні можна оптимально розмістити джерела світла;
- на основі державних будівельних норм було проведено аналіз стандартів освітлення приміщень на основі якого було визначено нормовані показники освітлення різних типів приміщень, в залежності від площини нормування освітленості та її висоти над рівнем підлоги, та від розряду та підрозряду зорової роботи;
- було проведено аналіз методів оптимізації, в якому були розглянуті:
 - локальні та глобальні методи відповідно до задач оптимізації;
 - прямі методи, методи першого та другого порядків, які поділяються за вимогами до гладкості й наявності в цільовій функції частинних похідних, аналітичні, чисельні методи та графічні методи;
- розглянуто ділення методів оптимізації на класи за природою множини X , а також за видом цільової функції та її обмежень;
- розглянуто методи параметричного, динамічного та стохастичного програмувань;
- розглянуто математичні моделі задач лінійного, цілочислового та нелінійного програмування;
- на підставі аналітичної інформації, що наведена в розділі 2, були розроблені способи оптимального розташування джерел штучного та природного освітлення;
- визначено основну вимогу до розроблення інноваційного програмного продукту – забезпечити можливість змодельовати приміщення з оптимально розташованими джерелами світла у ньому, а також надати можливість інтеграції з наявними 3D-редакторами моделей та підтримку основних ОС, такі як Windows, Linux та MacOS та його можливості:

- вводити вхідні дані, які застосовуються для розрахунку в графічному інтерфейсі;
 - перевіряти рівень освітлення в приміщенні;
 - проводити оптимізацію розміщення джерел світла;
 - отримувати повідомлення про відповідність до норм освітлення; отримувати повідомлення про можливість та необхідності оптимізації;
- визначено нефункціональні вимоги до продукту;
 - проведено аналіз можливостей програмного забезпечення для 3D-моделювання, який дозволив зробити висновки щодо доцільності створення доповнення до 3D-редактора та зробити вибір відповідного програмного продукту, переможцем став 3D-редактор Blender;
 - визначено структуру інноваційного програмного продукту, де зазначено, що програмний продукт є розширенням редактора тривимірних моделей Blender, що написаний мовою Python з використанням Blender Python API;
 - визначено архітектуру інноваційного програмного продукту та виділено наступні модулі: введення даних, оптимізації, перевірки рівня освітлення та розміщення 3D-об'єктів;
 - розглянуто алгоритми: введення даних, розрахунку рівня освітлення, оптимального розташування джерел світла;
 - визначено об'єкт та головну мету тестування;
 - визначено обсяг тестування, який містить:
 - перевірку вимог до інноваційного програмного продукту;
 - перевірку складу програмного продукту;
 - перевірку процесу встановлення програмного забезпечення;
 - перевірку роботи основних функцій програмного продукту;
 - визначено обмеження в умовах проведення тестування, зокрема це:

- відсутність вимог на заміну однієї програмної компоненти іншою, з аналогічними або розширеними функціями;
- відсутність вимог на заміну складу вхідних даних, зміну вхідних та вихідних документів;
- повторення окремих етапів випробувань не більше трьох разів;
- визначено послідовність виконання тестування;
- налаштовано перевірку роботи основних компонент програмного забезпечення, яка проводиться шляхом покриття юніт-тестами цих самих модулів за допомогою вбудованої в Python бібліотеки unittest, а також за допомогою тестових сценаріїв;
- створено тестові моделі та визначено тестові сценарії, які були відпрацьовані на діючому інноваційному програмному продукті;
- перевірено працездатність програмного продукту та його основних функцій відповідно до вимог;
- виявлено, що спосіб оптимального розміщення джерел світла реалізований у програмному забезпеченні у повному обсязі, програмний продукт працював у штатному режимі, збої та відмови в роботі ПЗ не виявлено;
- визначено актуальність проблеми, яку вирішує інноваційний програмний продукт, та виявлено проблеми, що виникають через відсутність програмного продукту: відсутність оцінки ергономічності приміщення на етапі проектування, відсутність оцінки “комфортабельності” приміщення тощо;
- створено діаграму ієрархії проблем у вигляді “дерева”;
- визначені групи зацікавлених осіб, їх інтерес та вплив, а також стратегії їх приваблення до інноваційного програмного продукту;
- визначено основні характеристики програмного рішення, зокрема визначено, що даний програмний продукт є по суті надбудовою над 3D-редактором, тому він повинен легко вбудовуватись з існуючими рішеннями 3D-моделювання;

- виявлені конкурентні переваги рішення:
 - можливість оцінити ергономічність прототипу в сенсі освітлення;
 - можливість оцінити “комфортабельність” приміщення;
 - зменшення споживання електроенергії;
 - зменшення кількості ламп для штучного освітлення завдяки їх оптимальному розміщенню;
- визначено, що основними клієнтами даного програмного продукту є забудовники та проектувальники;
- розроблено унікальну ціннісну пропозицію, розраховано доходи та витрати;
- розроблено бізнес-модель, де визначені: споживачі, проблема, рішення, унікальна ціннісна пропозиція, потоки доходів, структура витрат, прихована перевага, канали, ключові метрики;
- проведено аналіз бізнес-моделі, який дозволяє зробити висновок, що запропонований проект, який реалізує описаний у дисертації метод оптимального розміщення джерел світла, має перспективи у своїй подальшій реалізації.

Для реалізації програмного продукту був обраний 3D-редактор Blender, оскільки він є безкоштовним та має програмний інтерфейс для розширення інтерфейсу користувача, а також для додавання власних інструментів, наприклад автоматичного розміщення 3D-об’єктів на сцені. Тобто даний продукт є доповненням до редактора. Окрім цього, даний редактор є крос-платформним, що робить його доступним для користувачів різних популярних ОС.

Для реалізації способу оптимального розміщення джерел світла та розрахунку рівня освітлення було сформовано та реалізовано відповідні алгоритми.

Для забезпечення відповідності розробленого програмного забезпечення вимогам було розроблено методику тестування, створено

набори тестових моделей та послідовність дій. Виконано аналіз отриманих результатів та зроблено висновки щодо вдосконалення продукту.

Крім того, було розглянуто можливість створення стартап-проекту на основі програмної реалізації розробленого способу. Для досягнення цієї мети було проаналізовано існуючі проблеми, які виникають через неоптимальне розміщення джерел світла, та було виявлено зацікавлених сторін. Ними стали проектувальники та забудовники. Було запропоновано комерційне рішення та розглянуто його конкурентні переваги. Наведена бізнес-модель дозволяє стверджувати про комерційну життєздатність ідеї такого стартап-проекту.

СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Peter Raynham. The SLL Code for Lighting [Text] / P. Raynham, J. Fitzpatrick, P.R. Boyce // The Society of Light and Lighting, London. – 2012. – 340 pp. – ISBN 978-1-906846-21-3.
2. Peter Raynham. The SLL Lighting Handbook [Text] / P. Raynham, P.R. Boyce // The Society of Light and Lighting, London. – 2009. – 303 pp. – ISBN 978-1-906846-02-2.
3. Ir. Dr. Sam C. M. Hui. Lighting Calculations [Електронний ресурс] / Ir. Dr. Sam C. M. Hui // Faculty of Science and Technology. – 2017. – Режим доступу: http://ibse.hk/SBS5312/SBS5312_1718_05-lighting_calculations.pdf.
4. Mark S. Rea. IESNA Lighting Handbook 9th Edition [Text] / Mark S. Rea // Illuminating Engineering Society of North America. – 2000. – 1037 pp. – ISBN 0-87995-150-8.
5. Природне та штучне освітлення. ДБН В.2.5-28:2018 [Електронний ресурс] // Державне підприємство "Укрархбудінформ". – 2018. – Режим доступу: <http://www.minregion.gov.ua/wp-content/uploads/2018/12/V2528-1.pdf>.
6. Maya Platform Technologies [Електронний ресурс] // Autodesk Inc. – 2019. – Режим доступу: <http://www.autodesk.com/developer-network/platform-technologies/maya>.
7. Features of Maya [Електронний ресурс] // Autodesk Inc. – 2019. – Режим доступу: <https://www.autodesk.com/products/maya/features>.
8. 3ds Max Platform Technologies [Електронний ресурс] // Autodesk Inc. – 2019. – Режим доступу: <https://www.autodesk.com/developer-network/platform-technologies/3ds-max>.
9. Revit Platform Technologies [Електронний ресурс] // Autodesk Inc. – 2019. – Режим доступу: <https://www.autodesk.com/developer-network/platform-technologies/revit>.
10. AutoCAD Platform Technologies [Електронний ресурс] // Autodesk

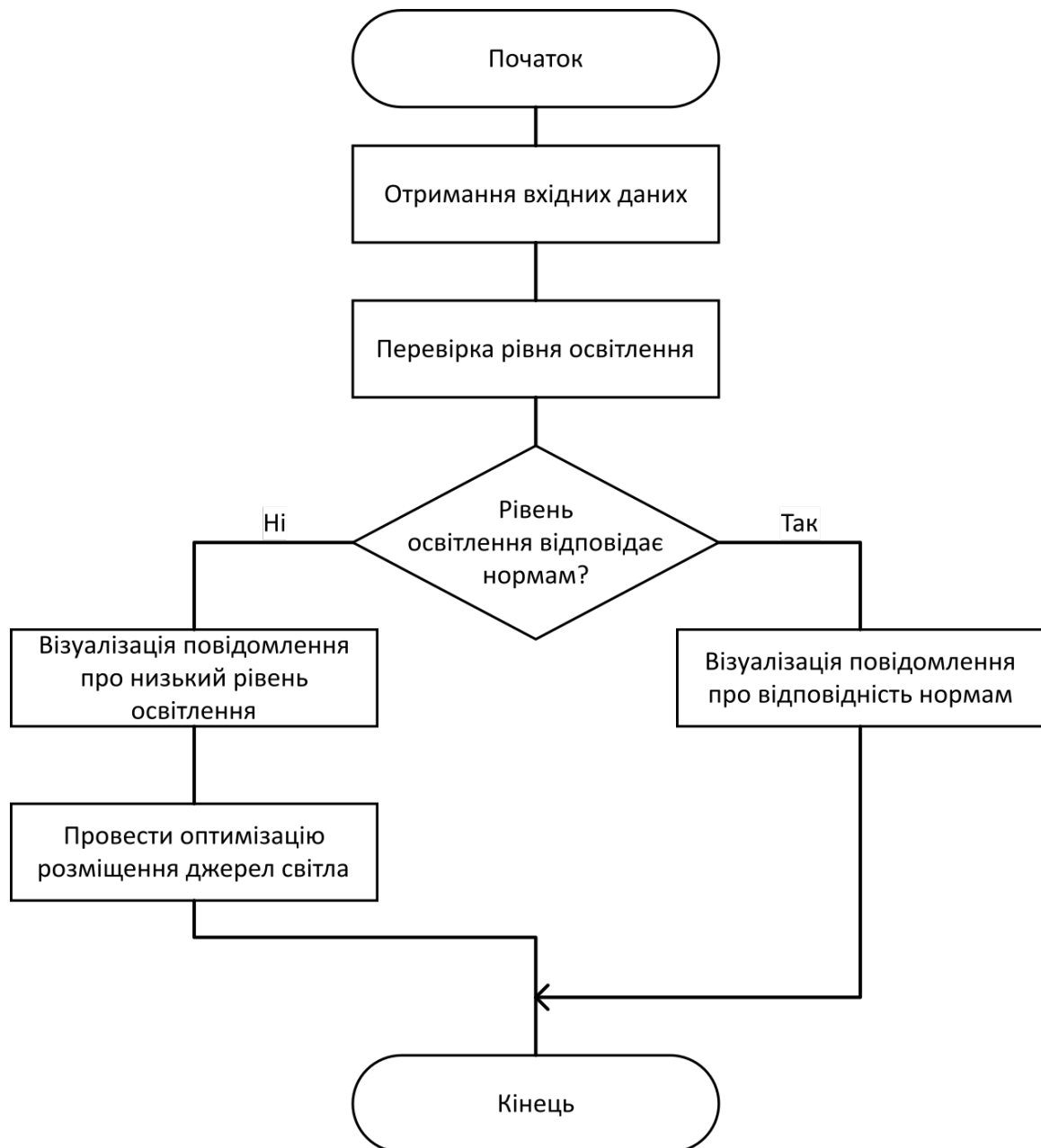
- Inc. – 2019. – Режим доступа: <https://www.autodesk.com/developer-network/platform-technologies/autocad>.
11. Autodesk Products [Электронный ресурс] // Autodesk Inc. – 2019. – Режим доступа: <https://www.autodesk.com/products>.
 12. Blender [Электронный ресурс] // Blender Foundation. – 2002. – Режим доступа: <https://www.blender.org/>.
 13. Цены на ПО для моделирования в 3D [Электронный ресурс] // Trimble Inc. – 2019. – Режим доступа: <https://www.sketchup.com/ru/plans-and-pricing#for-personal>.
 14. SketchUp C API [Электронный ресурс] // Trimble Inc. – 2019. – Режим доступа:
http://extensions.sketchup.com/developer_center/sketchup_c_api/sketchup/index.html.
 15. Blender Python API [Электронный ресурс] // Blender Foundation. – 2019. – Режим доступа:
https://docs.blender.org/api/current/info_quickstart.html.

ДОДАТКИ

Додаток 1

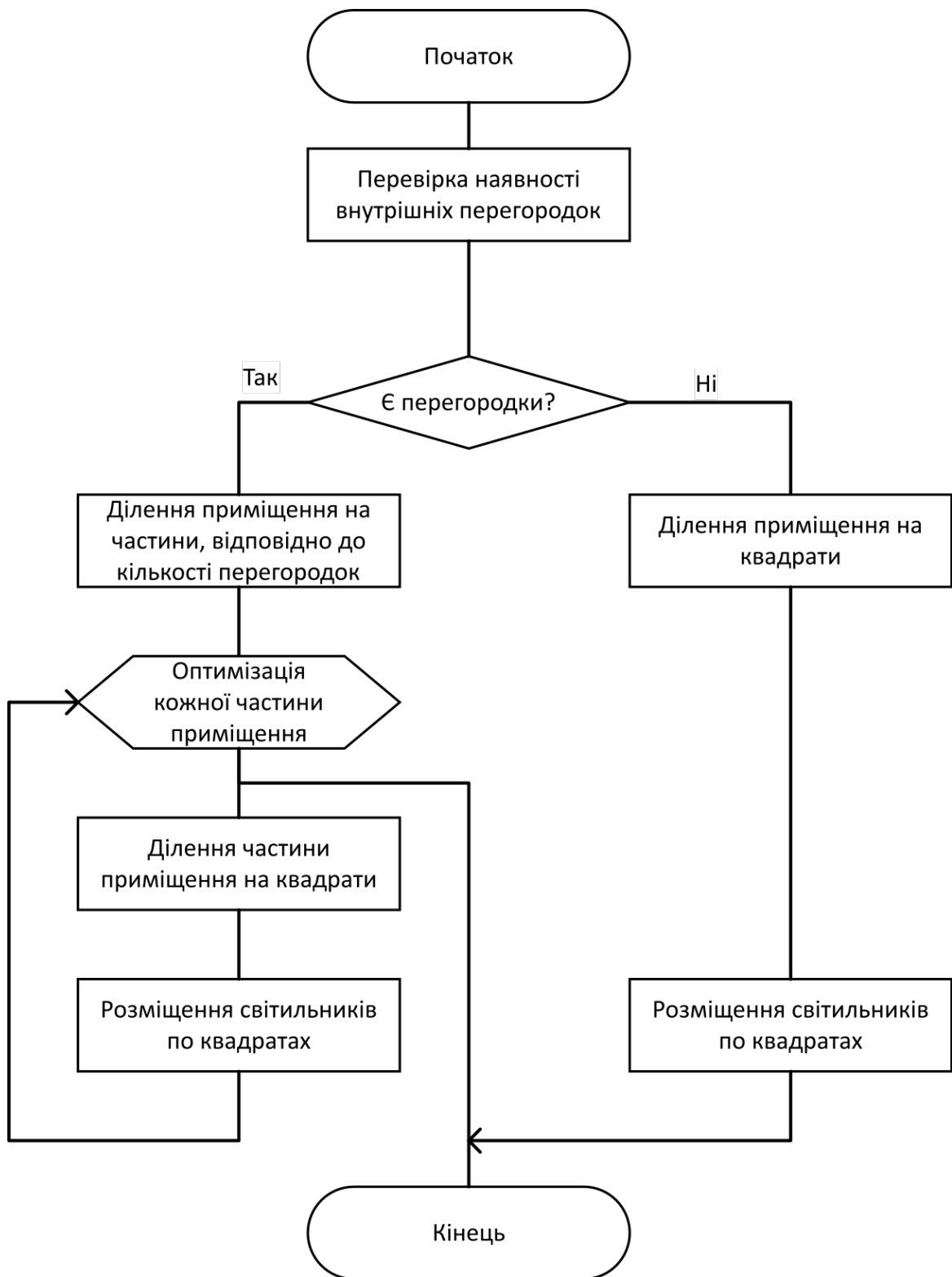
Копії графічних матеріалів

Алгоритм розрахунку рівня освітлення



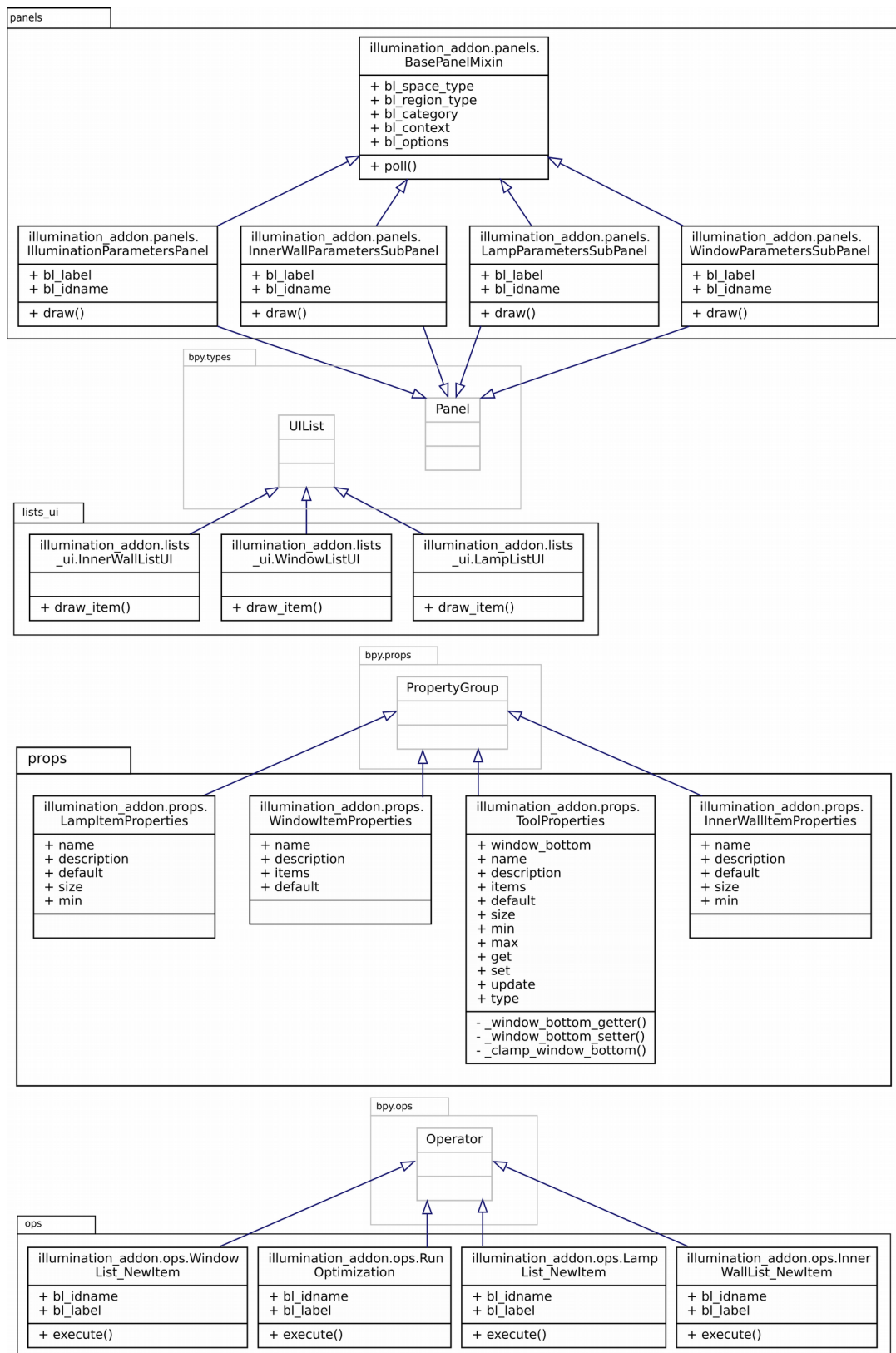
Метод та програмне забезпечення оптимального розміщення джерел світла в приміщенні Глінська Євгенія Михайлівна, КП-81мп

Алгоритм оптимального розміщення джерел світла



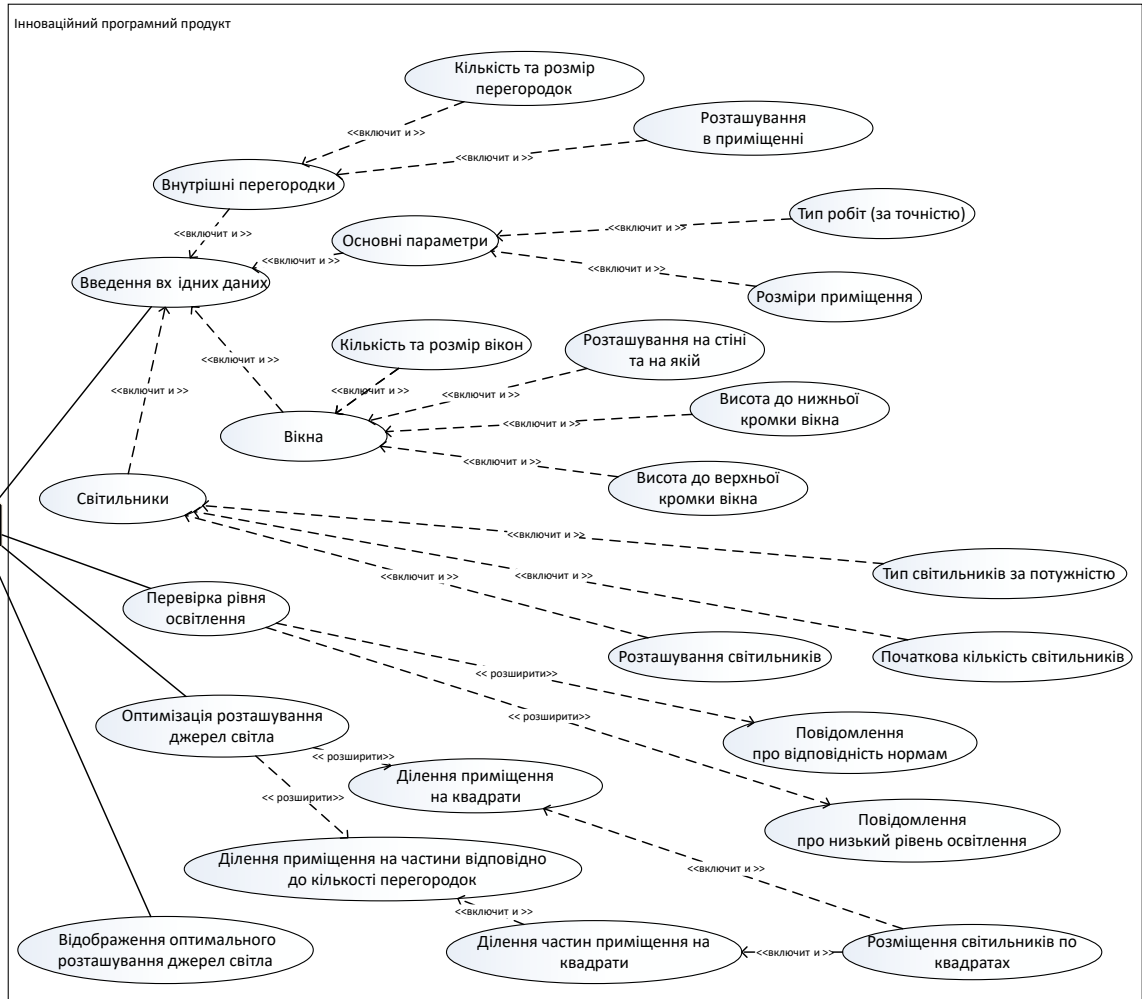
Метод та програмне забезпечення оптимального розміщення джерел світла в приміщенні Глінська Євгенія Михайлівна, КП-81мп

Діаграма класів графічного інтерфейсу



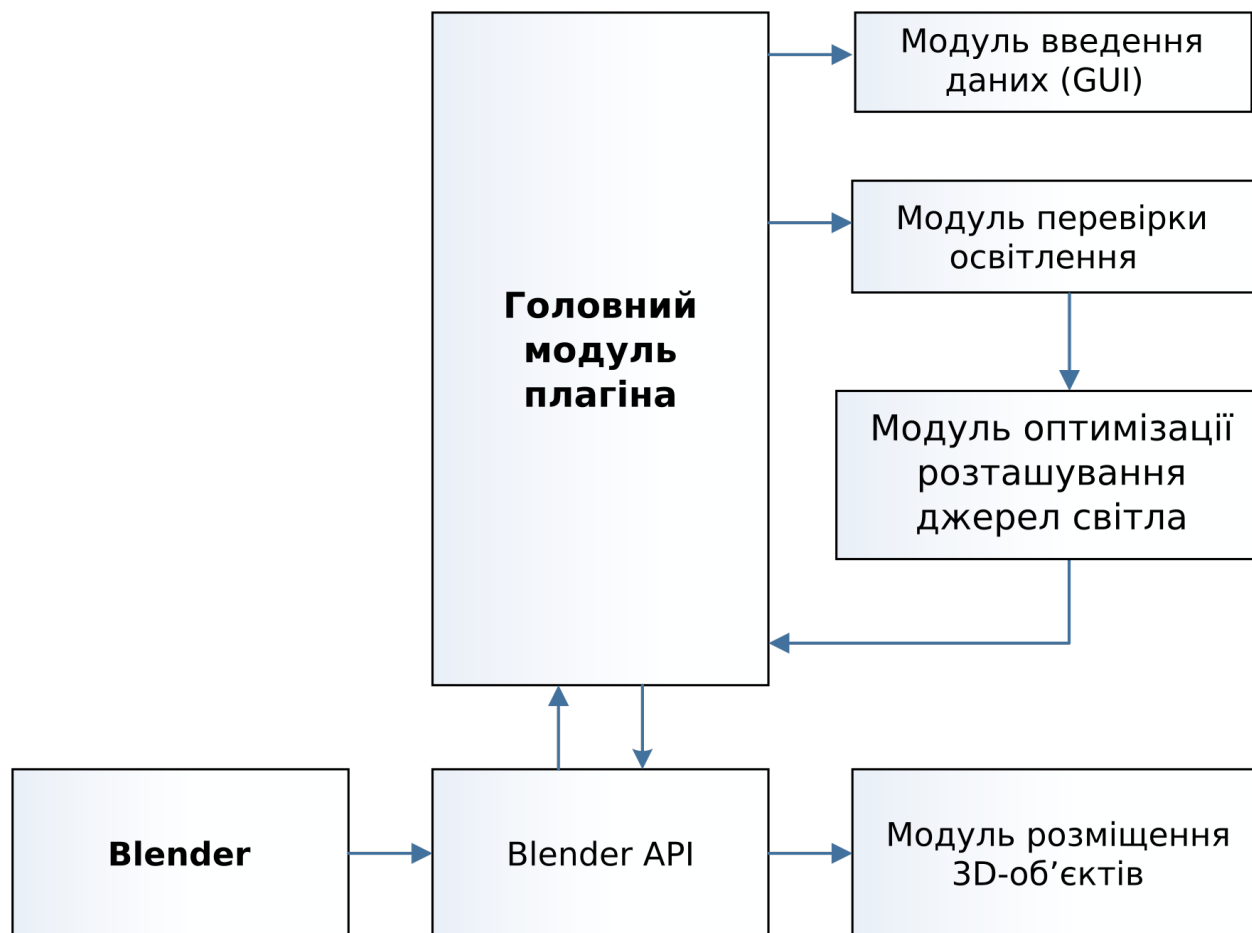
Метод та програмне забезпечення оптимального розміщення джерел світла в приміщенні
Глінська Євгенія Михайлівна, КП-81мп

Діаграма варіантів використання



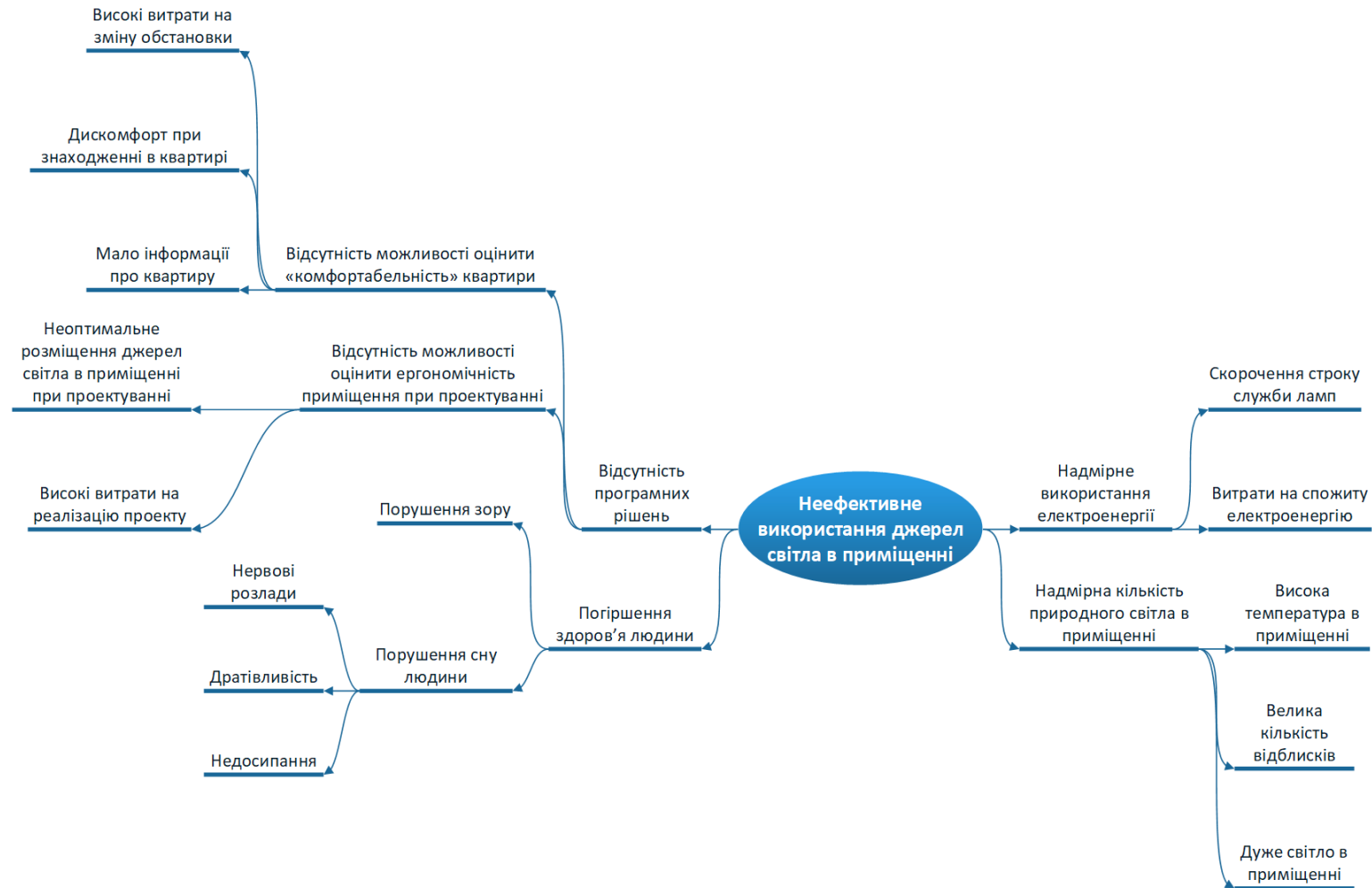
Метод та програмне забезпечення оптимального розміщення джерел світла в приміщенні
Глінська Євгенія Михайлівна, КП-81мп

Архітектура програмного продукту



Метод та програмне забезпечення оптимального розміщення джерел світла в приміщенні
Глінська Євгенія Михайлівна, КП-81мп

Дерево проблем



Метод та програмне забезпечення оптимального розміщення джерел світла в приміщенні
Глінська Євгенія Михайлівна, КП-81мп

Додаток 2

Лістинги тексту програми

Лістинг 1. Фрагменти коду модулів обчислення рівня світла та оптимізації розміщення світильників

```
import input, scenario_night, scenario_day, geometry
from typing import List, Tuple
from constants import *

def optimize(
    work: Work,
    width: float,
    depth: float,
    light: Light,
    light_height: float,
    light_positions: List[Tuple[float, float]],
    w_top_height: float,
    w_bottom_height: float,
    front_windows: List[Tuple[float, float]],
    left_windows: List[Tuple[float, float]],
    right_windows: List[Tuple[float, float]],
    partitions: List[float]):
    error = input.check_input_errors(
        work,
        width,
        depth,
        light,
        light_height,
        light_positions,
        w_top_height,
        w_bottom_height,
        front_windows,
        left_windows,
        right_windows,
        partitions)
    if error:
        raise Exception(error)

    def _generate():
        for room_left, room_front, room_width, room_depth in
            geometry.get_divided_rooms(width, depth, partitions):
                yield __optimize(
                    room_left,
                    room_front,
```

```

        room_width,
        room_depth,
        work,
        width,
        depth,
        light,
        light_height,
        light_positions,
        w_top_height,
        w_bottom_height,
        front_windows,
        left_windows,
        right_windows)

    return list(_generate())

def __optimize(
    room_left: float,
    room_front: float,
    room_width: float,
    room_depth: float,
    work: Work,
    width: float,
    depth: float,
    light: Light,
    light_height: float,
    light_positions: List[Tuple[float, float]],
    w_top_height: float,
    w_bottom_height: float,
    front_windows: List[Tuple[float, float]],
    left_windows: List[Tuple[float, float]],
    right_windows: List[Tuple[float, float]]):

    print(
        'Room optimization: {0} m, {1} m - {2} m x {3}
m:'.format(room_left, room_front, room_width, room_depth))
    room_light_positions = list(
        geometry.get_divided_lights(room_left, room_front, room_width,
room_depth, light_positions))
    room_front_windows = list(
        geometry.get_divided_windows(room_left, room_width,
front_windows)) if room_front == 0 else None
    room_left_windows = list(geometry.get_divided_windows(room_front,

```

```

room_depth,

left_windows)) if room_left == 0 and \

left_windows is not None else None
    room_right_windows = list(geometry.get_divided_windows(room_front,
room_depth,

right_windows)) if room_left + room_width == width \

and right_windows is not None else None

    light_optimization_result = scenario_night.optimize(work,
room_width, room_depth, light, light_height,

room_light_positions)

    if not light_optimization_result:
        print('The lamps optimization is unnecessary')
        new_light, new_room_light_positions = light_optimization_result
        if not new_light:
            print('Optimal positions of {0}
lamps:'.format(len(new_room_light_positions)))
        else:
            print('Optimal positions of {0} lamps with power
{1}:'.format(len(new_room_light_positions),

new_light.text))
            for w, d in new_room_light_positions:
                print(' - distance in width: {0} m; depth: {1} m'.format(w
+ room_left, d + room_front))

        if room_front_windows is None and room_left_windows is None and
room_right_windows is None:
            print('Windows optimization is impossible')
        else:
            window_optimization_result = scenario_day.optimize(work,
room_width, room_depth, w_top_height,

w_bottom_height, room_front_windows, room_left_windows,

room_right_windows)
            if not window_optimization_result:

```



```

        print('Windows optimization is unnecessary')
    else:
        w_width, w_height, front_ws, left_ws, right_ws =
window_optimization_result
        print('Optimal positions of windows {0} x {1}
m:'.format(w_width, w_height))
        if front_ws is not None and len(front_ws) > 0:
            print(' - frontals:')
            for w in front_ws:
                print(' - distance in width: {0} m'.format(w +
room_left))
        if left_ws is not None and len(left_ws) > 0:
            print(' - lefts:')
            for w in left_ws:
                print(' - distance in depth: {0} m'.format(w +
room_front))
        if right_ws is not None and len(right_ws) > 0:
            print(' - rights:')
            for w in right_ws:
                print(' - distance in depth: {0} m'.format(w +
room_front))
        return (
            __get_light_results(room_left, room_front, light,
room_light_positions, light_optimization_result),
            __get_window_results(room_left, room_front, w_top_height -
w_bottom_height, room_front_windows,
                                room_left_windows, room_right_windows,
window_optimization_result))

def __get_light_results(room_left, room_front, light, light_positions,
light_results):
    if light_results is None:
        return light, [(w + room_left, d + room_front) for w, d in
light_positions]
    new_light, new_light_positions = light_results
    return (light if new_light is None else new_light,
            [(w + room_left, d + room_front) for w, d in
new_light_positions])

def __get_window_results(room_left, room_front, window_height,
room_front_windows, room_left_windows,
                        room_right_windows, window_result):
    if window_result is None:

```

```
        return (window_height,
                [(p + room_left, w) for p, w in room_front_windows] if
room_front_windows is not None else None,
                [(p + room_front, w) for p, w in room_left_windows] if
room_left_windows is not None else None,
                [(p + room_front, w) for p, w in room_right_windows] if
room_right_windows is not None else None)
        w_width, w_height, front_ws, left_ws, right_ws = window_result
        return (w_height,
                [(p + room_left, w_width) for p in front_ws] if front_ws is
not None else None,
                [(p + room_front, w_width) for p in left_ws] if left_ws is
not None else None,
                [(p + room_front, w_width) for p in right_ws] if right_ws
is not None else None)
```

Лістинг 2. Фрагмент коду модуля введення даних

```
import bpy.types

from .props import WindowItemProperties, ToolProperties,
InnerWallItemProperties, LampItemProperties
from .lightcalc import optimization, constants

class WindowList_NewItem(bpy.types.Operator):
    """ Add a new item to the list """

    bl_idname = "windows.new_item"
    bl_label = "Add a new window"

    def execute(self, context):
        tool = context.scene.my_tool
        p: WindowItemProperties = tool.windows.add()
        template: WindowItemProperties = tool.window
        p.height = template.height
        p.width = template.width
        p.position = template.position
        p.wall = template.wall

        return {'FINISHED'}

class WindowList_DeleteItem(bpy.types.Operator):
    """ Delete item from the list """

    bl_idname = "windows.delete_item"
    bl_label = "Delete a window"

    def execute(self, context):
        tool: ToolProperties = context.scene.my_tool
        windows: WindowItemProperties = tool.windows
        windows.remove(tool.selected_window_index)

        return {'FINISHED'}

class InnerWallList_NewItem(bpy.types.Operator):
    """ Add a new item to the list """
```

```

bl_idname = "inner_walls.new_item"
bl_label = "Add a new inner wall"

def execute(self, context):
    tool: ToolProperties = context.scene.my_tool
    p: InnerWallItemProperties = tool.inner_walls.add()
    template: InnerWallItemProperties = tool.inner_wall
    p.size = template.size
    p.position = template.position

    return {'FINISHED'}

class InnerWallList_DeleteItem(bpy.types.Operator):
    """ Delete item from the list """

    bl_idname = "inner_walls.delete_item"
    bl_label = "Delete an inner wall"

    def execute(self, context):
        tool: ToolProperties = context.scene.my_tool
        inner_walls: InnerWallItemProperties = tool.inner_walls
        inner_walls.remove(tool.selected_inner_wall_index)

        return {'FINISHED'}

class LampList_NewItem(bpy.types.Operator):
    """ Add a new item to the list """

    bl_idname = "lamps.new_item"
    bl_label = "Add a new lamp"

    def execute(self, context):
        tool: ToolProperties = context.scene.my_tool
        p: LampItemProperties = tool.lamps.add()
        template: LampItemProperties = tool.lamp
        p.position = template.position

        return {'FINISHED'}

class LampList_DeleteItem(bpy.types.Operator):

```

```

""" Delete item from the list """

bl_idname = "lamps.delete_item"
bl_label = "Delete a lamp"

def execute(self, context):
    tool: ToolProperties = context.scene.my_tool
    lamps: LampItemProperties = tool.lamps
    lamps.remove(tool.selected_lamp_index)

    return {'FINISHED'}

class RunOptimizationButton(bpy.types.Operator):
    bl_label = "Run optimization"
    bl_idname = "bt.run_optimization"

    def execute(self, context):
        tool: ToolProperties = context.scene.my_tool

        try:
            windows = {
                'front_windows': [],
                'left_windows': [],
                'right_windows': []
            }
            w: WindowItemProperties
            for w in tool.windows:
                key = w.wall + '_windows'
                windows[key].append((float(w.width), w.position))

            optimization.optimize(
                getattr(constants.Works, tool.work_class),
                tool.room_size[0],
                tool.room_size[1],
                getattr(constants.Lights, tool.lamps_power),
                tool.light_elevation,
                [(l.position[0], l.position[1]) for l in tool.lamps],
                tool.window_top,
                tool.window_bottom,
                **windows,
                partitions=[])

```

```
except Exception as error:
    print(error)
```

```
return {'FINISHED'}
```

```
class WindowListUI(bpy.types.UIList):
    def draw_item(self, context, layout, data, item: WindowItemProperties,
icon, active_data, active_propname, index):
        if self.layout_type in {'DEFAULT', 'COMPACT'}:
            row = layout.row(align=True)
            row.label(icon='DRIVER_DISTANCE')
            row.prop(item, 'width', text='', emboss=False)
            row.prop(item, 'height', text='', emboss=False)
            row.label(icon='CUBE')
            row.prop(item, 'wall', text='', emboss=False)
            row.label(icon='FORWARD')
            row.prop(item, 'position', text='', emboss=False)

        elif self.layout_type in {'GRID'}:
            layout.alignment = 'CENTER'
            layout.label("", icon='OBJECT_DATAMODE')
```

```
class InnerWallListUI(bpy.types.UIList):
    def draw_item(self, context, layout, data, item:
InnerWallItemProperties, icon, active_data, active_propname, index):
        if self.layout_type in {'DEFAULT', 'COMPACT'}:
            row = layout.row(align=True)
            row.label(icon='ARROW_LEFTRIGHT')
            row.prop(item, 'size', text='', emboss=False)
            row.label(icon='TRACKER')
            row.prop(item, 'position', text='', emboss=False)

        elif self.layout_type in {'GRID'}:
            layout.alignment = 'CENTER'
            layout.label("", icon='OBJECT_DATAMODE')
```

```
class LampListUI(bpy.types.UIList):
    def draw_item(self, context, layout, data, item:
InnerWallItemProperties, icon, active_data, active_propname, index):
        if self.layout_type in {'DEFAULT', 'COMPACT'}:
            row = layout.row(align=True)
            row.label(icon='LIGHT')
```

```

        row.label(icon='TRACKER')
        row.prop(item, 'position', text='', emboss=False)

    elif self.layout_type in {'GRID'}:
        layout.alignment = 'CENTER'
        layout.label("", icon='LIGHT')

class WindowItemProperties(PropertyGroup):
    width: EnumProperty(
        name="Window(s) width (in centimeters)",
        description="Choose the window(s) width",
        items=[
            ("0.6", "60 cm", ""),
            ("0.9", "90 cm", ""),
            ("1.00", "100 cm", ""),
            ("1.20", "120 cm", ""),
            ("1.35", "135 cm", ""),
            ("1.50", "150 cm", ""),
            ("1.80", "180 cm", "")
        ]
    )

    height: EnumProperty(
        name="Window(s) height (in centimeters)",
        description="Choose the window(s) height",
        items=[
            ("0.6", "60 cm", ""),
            ("0.9", "90 cm", ""),
            ("1.20", "120 cm", ""),
            ("1.35", "135 cm", ""),
            ("1.50", "150 cm", ""),
            ("1.80", "180 cm", "")
        ]
    )

    position: FloatProperty(
        name="Window wall position",
        description="Specify window location on the wall in meters",
        default=0
    )

    wall: EnumProperty(
        name="Wall",

```

```

        description="Choose wall of the window",
        items=[
            ("front", "Front", ""),
            ("left", "Left", ""),
            ("right", "Right", "")
        ]
    )

class InnerWallItemProperties(PropertyGroup):
    position: FloatVectorProperty(
        name="Wall position",
        description="Specify inner wall location",
        default=[0.0, 0.0],
        size=2,
        min=0
    )

    size: FloatVectorProperty(
        name="Wall size",
        description="Specify inner wall size",
        default=[0.5, 0.5],
        size=2,
        min=.5
    )

class LampItemProperties(PropertyGroup):
    position: FloatVectorProperty(
        name="Lamp position",
        description="Specify lamp location",
        default=[0.0, 0.0],
        size=2,
        min=0
    )

class ToolProperties(PropertyGroup):
    work_class: EnumProperty(
        name="Work precision",
        description="Select work's precision",
        items=[
            ("high_precision", "High", ""),
            ("normal_precision", "Normal", ""),
            ("low_precision", "Low", ""),
            ("draft_precision", "Draft", "")
        ]
    )

```



```

    ]
)

room_size: FloatVectorProperty(
    name="Room size",
    description="Dimensions: width and length in meters",
    default=[1.0, 1.0],
    size=2,
    min=1.0,
    max=50.0
)

light_elevation: FloatProperty(
    name="Lights' elevation",
    description="Distance from floor to the lights in meters",
    default=2.0,
    min=0.0,
    max=50.0
)

def _window_bottom_getter(self):
    return self.get('window_bottom', 1.0)

def _window_bottom_setter(self, value):
    self['window_bottom'] = min(self.window_top, value)

window_bottom: FloatProperty(
    name="Window bottom elevation",
    description="Distance from floor to the bottom of the windows in
meters",
    default=1.0,
    min=0.0,
    max=50.0,
    get=_window_bottom_getter,
    set=_window_bottom_setter
)

def _clamp_window_bottom(self: 'ToolProperties', context):
    self.window_bottom = min(self.window_top, self.window_bottom)

window_top: FloatProperty(
    name="Window top elevation",
    description="Distance from floor to the top of the windows in

```

```

meters",
    default=2.0,
    min=0.0,
    max=50.0,
    update=_clamp_window_bottom
)

inner_walls: CollectionProperty(type=InnerWallItemProperties)
inner_wall: PointerProperty(type=InnerWallItemProperties)
selected_inner_wall_index: IntProperty(name="Index for inner walls",
default=-1)

windows: CollectionProperty(type=WindowItemProperties)
window: PointerProperty(type=WindowItemProperties)
selected_window_index: IntProperty(name="Index for windows", default=-
1)

lamps: CollectionProperty(type=LampItemProperties)
lamp: PointerProperty(type=LampItemProperties)
selected_lamp_index: IntProperty(name="Index for lamps", default=-1)

lamps_power: EnumProperty(
    name="Lamp type",
    description="Select lamp type",
    items=[
        ("T8_36_Watts", "T8: 36 W", ""),
        ("T5_18_Watts", "T5: 18 W", ""),
        ("B_100_Watts", "100 W", ""),
        ("B_60_Watts", "60 W", ""),
        ("B_40_Watts", "40 W", "")
    ]
)

```

Додаток 3

Копія слайдів презентації

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО”



ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ
КАФЕДРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРНИХ
СИСТЕМ

Метод та програмне забезпечення оптимального розміщення джерел світла в приміщенні

Виконала: Глінська Євгенія Михайлівна

Науковий керівник: к.т.н., доцент Сулема Євгенія Станіславівна

Київ – 2019



Актуальність дослідження

- Надмірне використання електроенергії
- Погіршення здоров'я людини
- Надмірна кількість природного світла в приміщенні
- Відсутність програмних рішень



Науково-інноваційна задача

Об'єкт дослідження: процес оптимізації розміщення джерел світла.

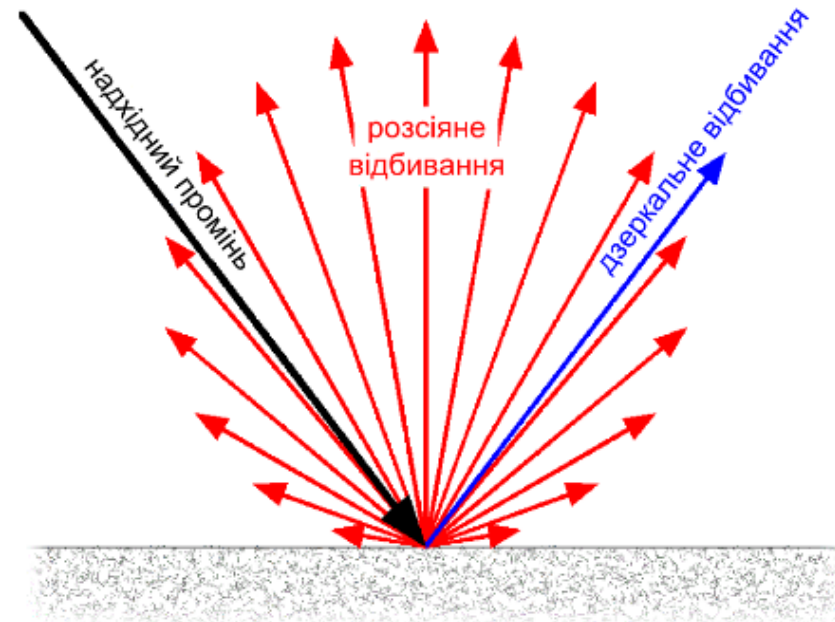
Предмет дослідження: способи та алгоритми оптимального розміщення джерел світла.

Наукове завдання: розробити спосіб оптимального розміщення джерел світла в приміщенні.

Мета дослідження: створення програмного забезпечення, яке дозволить оптимально розмістити джерела світла з метою підвищення ергономічності приміщення в сенсі світла та зменшення споживання електроенергії

Типи світла та його відбиття

- Прямий
- Непрямий
 - Дифузне відбиття
 - Дзеркальне відбиття
- Комбінований





Методи для розрахунку рівня освітлення

$$E = \frac{N \cdot n \cdot F \cdot CU \cdot LLF}{A}$$

Метод люмену

$$E = \frac{\Phi}{A(1 - R_{av})} \quad E_{ind} = \frac{\Phi}{A} \cdot \frac{R_{av}}{1 - R_{av}}$$

Метод Сампнера

$$E = \frac{S_e \cdot t_0 \cdot t_1 \cdot r_1 \cdot 100}{m \cdot K_{б\text{уд}} \cdot K_з \cdot n_e \cdot S_n}$$

Формула для розрахунку
рівня КПО

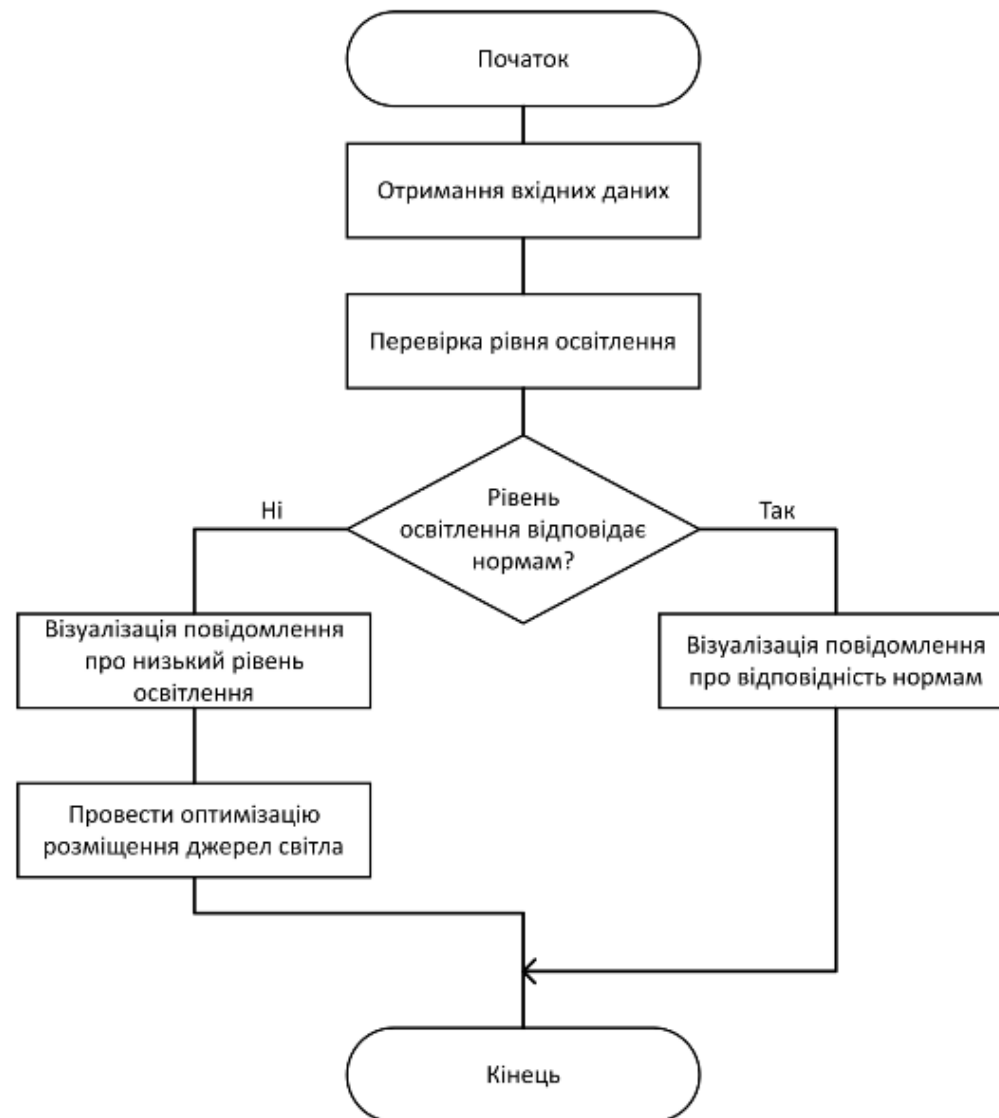
$$E = \frac{I}{D^2} \cdot \cos \alpha$$

Поточковий метод

Методи для розрахунку рівня освітлення

Назва	Тип освітлення	Додаткові розрахунки	Основа методу	Обмеження
Метод люмену	Прямий, непрямий	Коефіцієнт утилізації, параметр втрати світла	—	Необхідно додатково визначати коефіцієнт утилізації та параметр втрати світла
Метод Сампнера	Прямий, непрямий	Відбивні властивості поверхні, кількість світлового потоку, що входить в простір	Все світло, яке входить в простір, має або покинути його, або бути поглинутим однією з поверхонь простору	Мають бути відомі відбивні властивості поверхні та кількість світлового потоку, що увійшло в простір
Поточковий метод	Прямий	Інтенсивність джерела світла, відстань від джерела світла, орієнтація поверхні	Закон зворотніх квадратів, закон косинусів	Максимальний фіз. розмір поверхні має бути не більше ніж 1/5 висоти монтажу над точкою оцінки

Алгоритм розрахунку рівня освітлення

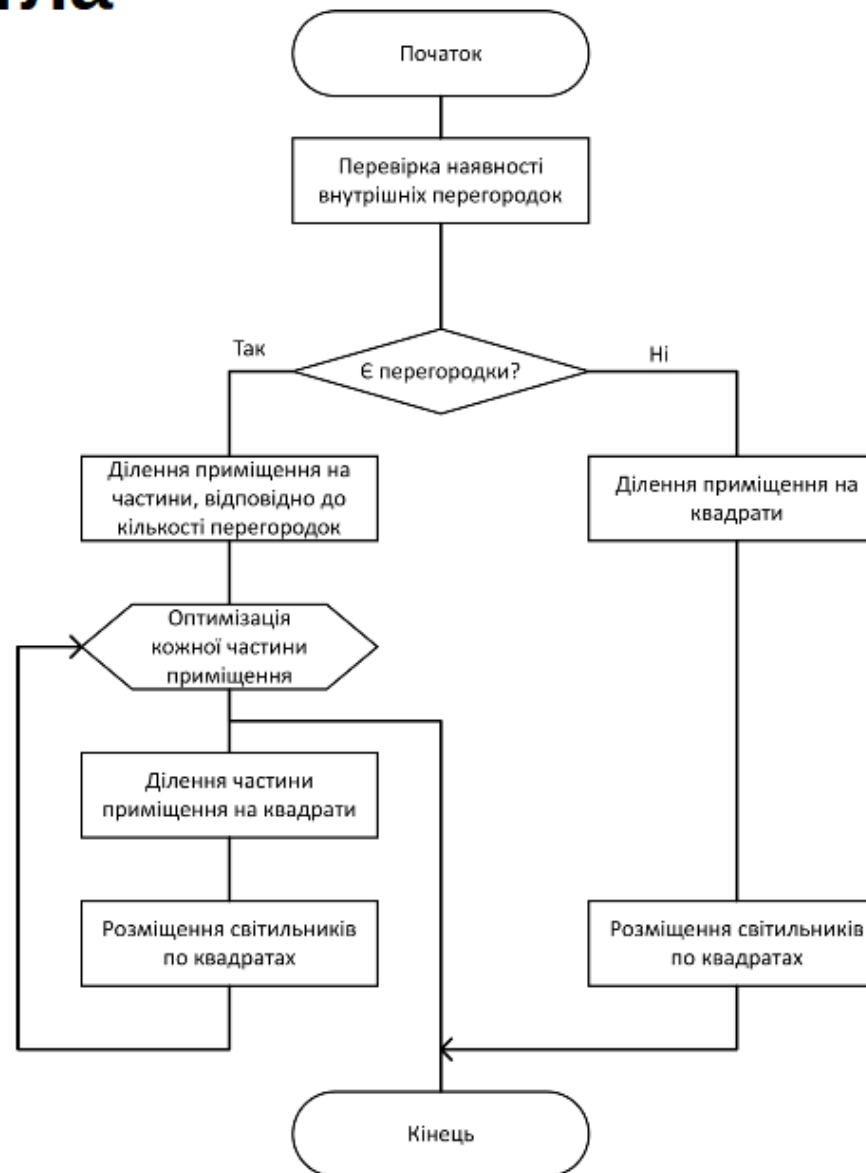




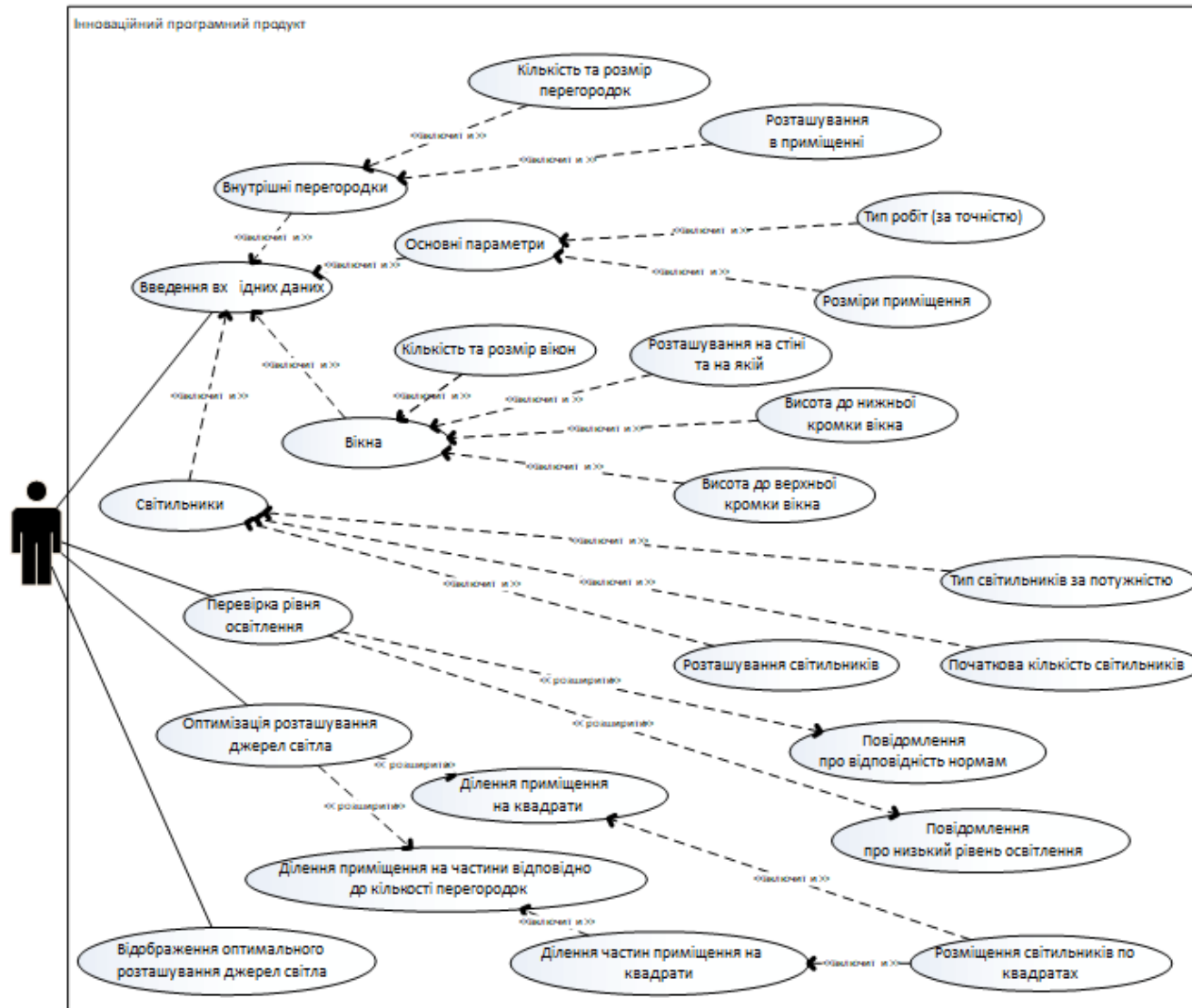
Норми освітлення

Приміщення	Норма освітлення, лк
Кабінети, робочі кімнати і офіси	300
Приміщення для роботи з дисплеями і відеотерміналами	200-400
Читальні зали	400
Аудиторії, навчальні кабінети	400
Житлові кімнати, вітальні, спальні	150
Кабінети, бібліотеки	300

Алгоритм оптимального розміщення джерел світла



Вимоги до ПЗ: функціональні





Вимоги до ПЗ: нефункціональні

- Графічний інтерфейс
- Інтеграція з 3D-редактором
- Виконання з середовища редактора, який забезпечує підтримку основних ОС – Windows, Linux, MacOS

Порівняння редакторів

	Autodesk 3ds Max, Revit, AutoCAD	SketchUp	Blender
Вартість ліцензії	\$1545 на рік і вище	\$299 на рік	\$0
Засоби розроблення та інтеграції власних рішень	Autodesk SDK	SketchUp SDK	Blender Python API
Мова програмування	Python, MEL, C+ +, C#, .NET	C/C++	Python
Крос-платформеність	Часткова	Часткова	Повна
Повнота документації для розробників	+/-	+/-	+/-
Можливість працювати з інструментами редактора	Невідомо	Можливо	Можливо



Обрані засоби розроблення

3D-редактор:

- Blender

Мова програмування:

- Python

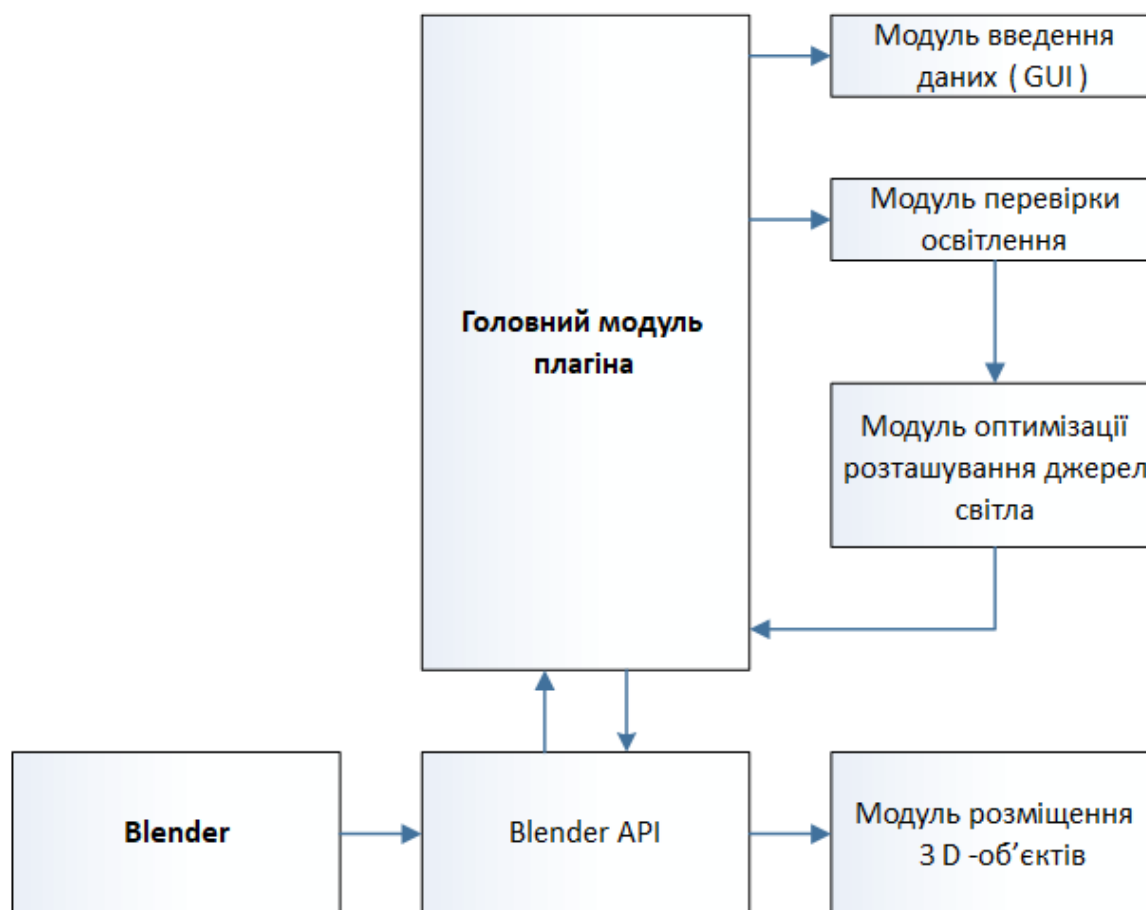
Контроль версій:

- Git

IDE:

- Pycharm Community Edition

Архітектура розробленого ПЗ



Бізнес-модель стартап-проекту

Проблема	Рішення	Унікальна ціннісна пропозиція	Прихована перевага	Споживачі
<ul style="list-style-type: none">▪ відсутність програмних рішень▪ відсутність оцінки ергономічності приміщення на етапі проектування▪ погіршення здоров'я людини▪ надмірне споживання електроенергії	програмний продукт, здатний оптимально розмістити джерела світла на етапі проектування	розроблений метод оптимального розміщення джерел світла з метою підвищення ергономічності приміщення в сенсі світла та зменшення споживання електроенергії	відповідність до норм освітлення, зазначених в ДБН України	Компанії, що займаються проектуванням та забудовою нових будівель
	Ключові показники		Канали	
	кількість проданих ліцензій		відділи інтеграції компаній-споживачів	
Структура витрат			Потоки доходів	
<ul style="list-style-type: none">▪ науково-дослідні та дослідно-конструкторські роботи▪ оплата праці персоналу▪ утримання робочих місць персоналу та виробничої інфраструктури▪ обладнання, ліцензії програмного забезпечення для розробки▪ податкові витрати			<ul style="list-style-type: none">▪ доходи від продажу ліцензій програмного забезпечення▪ надання підтримки програмного забезпечення	

Фінансовий план стартап-проекту



	Одиниці виміру: тис. доларів США				
Півріччя	Загальні витрати	Оплата праці	Підсумок витрат	Заплановані доходи	Прибуток (без оподаткування)
1	28	64	92	0	-92
2	27	73	100	265	165



Науково-інноваційна новизна

Розроблено програмне забезпечення, що дозволяє за допомогою розробленого способу оптимального розміщення джерел світла в приміщенні оптимально розмістити штучні та природні джерела світла. Перевагами даного способу є простота реалізації та висока ефективність за рахунок відсутності потреби будувати математичну модель та формувати великої кількості критеріїв та обмежень.

Суть методу оптимального розміщення джерел світла полягає у розбитті площі приміщення на квадрати, площа яких залежить від потужності обраного штучного джерела світла або кількості та розміру вікон бокового освітлення, які розраховуються на основі обчислення рівня освітлення приміщення за рівнем КПО.



Висновки

1. Розглянуто основні типи світла та його відбиття, методи розрахунку рівня освітлення, розроблено спосіб оптимального розміщення джерел світла на основі результатів аналізу існуючих методів оптимізації.
2. Виконано розроблення програмної реалізації способу з використанням 3D-редактора Blender та аналіз отриманих результатів.
3. Проведено оцінку комерційної життєздатності продукту: побудована бізнес-модель стартап-проєкту, розраховано прогнозовані показники потенційної прибутковості.



Апробування отриманих результатів

1. XII наукова конференція магістрантів та аспірантів
“Прикладна математика та комп’ютинг” ПМК-2019



Check ID:
1000790202

Check date:
17.12.2019 21:49:30 GMT+0

Check type:
Doc vs Internet + Library

Report date:
17.12.2019 21:51:37 GMT+0

User ID:
83363

File name: 2019_Master_Glinskaya_2

File ID: 1000800203 Page count: 80 Word count: 14220 Character count: 110980 File size: 1.10 MB

[illegible]

6.55% Internet Matches	176
------------------------	-----

Page 82

7.04% Library matches	320
-----------------------	-----

Page 84

Quotes	1
--------	---

Page 85

No references found

No exclusions found

Character replacement 11



Дякую за увагу